



# **Arquitectura de Entornos Web**

---

***Ciclo: 2018***



# ÍNDICE

Presentación	5
Red de contenidos	7
 <b>Unidad de aprendizaje 1</b>	
<b>Fundamentos de un Proyecto Web y Elementos Gráficos</b>	9
1.1 Tema 1 : Fundamentos de un proyecto web	11
1.1.1. : Generalidades	11
1.1.2. : Etapas de un proyecto web	25
1.2 Tema 2 : Elementos gráficos para la web	57
1.2.1. : Manejo de un editor de imágenes	57
1.2.2. : Creación de banners, menús y botones	60
 <b>Unidad de aprendizaje 2</b>	
<b>HTML5 Y CSS3</b>	73
2.1 Tema 3 : Estructura HTML y aplicación de estilos CSS	75
2.1.1. : Entorno del programa (Editor HTML)	76
2.1.2. : Creación de un sitio web (estructura de carpetas)	77
2.1.3. : Estructura de un documento HTML	81
2.2 Tema 4 : Estilos CSS	105
2.2.1. : Estilos CSS	107
2.2.1.1 : Tipos de estilos	109
2.2.1.2 : Estilos aplicados a etiquetas y de clase	111
 <b>Unidad de aprendizaje 3</b>	
<b>ESTRUCTURAS Y LAYOUT</b>	133
3.1 Tema 5 : Creación de bloques	135
3.1.1. : Etiquetas de bloques	135
3.1.1.1 : Estilos básicos del bloque	137
3.1.1.2 : Estilos para bloques	139
3.1.1.3 : Propiedad box-sizing	143
3.2 Tema 6 : Creación de Menu	167
3.2.1 : Estilos a vínculos: creación de menu	168

**Unidad de aprendizaje 4**

<b>MULTIMEDIA</b>	187
4.1 Tema 7 : Elementos multimedia	189
4.1.1. : Etiquetas AUDIO y VIDEO	189
4.1.2. : Codificar audio y video para la web	189
4.1.3. : Convertidor de audio y video online	192
4.1.4. : Etiqueta iframe: Mostrar video en Youtube	192
4.1.5. : Uso de la herramienta LightBox para la gestión de imágenes y video	193

**Unidad de aprendizaje 5**

<b>TRANSICIÓN Y ANIMACIÓN CON CSS</b>	213
5.1 Tema 8 : Transición y Animación	215
5.1.1 : Transición	215
5.1.2 : Animación	221

**Unidad de Aprendizaje 6**

<b>FORMULARIOS</b>	225
6.1 Tema 9 : Formularios	227
6.1.1 : Elementos clásicos de formulario	227
6.1.2 : Nuevos atributos para la etiqueta INPUT	231
6.1.3 : Atributos del formulario	233
6.2 Tema 10 : Publicación de un sitio web	243
6.2.1 : Uso de un sitio Web para el envío de datos por un formulario	245
6.2.2 : Publicación del proyecto web en un sitio web	249



# PRESENTACIÓN

La web 3.0 es la nueva evolución que han experimentado todos los navegadores modernos, los cuales, permiten marcar el contenido de forma semántica, para así poder ser indexado con mucha más fiabilidad y efectividad.

HTML5 es un nuevo lenguaje para el diseño de páginas web, que proporciona muchas más etiquetas de marcado de contenido entre los que destaca los bloques de contenidos, efectos multimedia, formularios, etc; y trabaja con lenguajes de programación JavaScript para elaborar páginas mas interactivas.

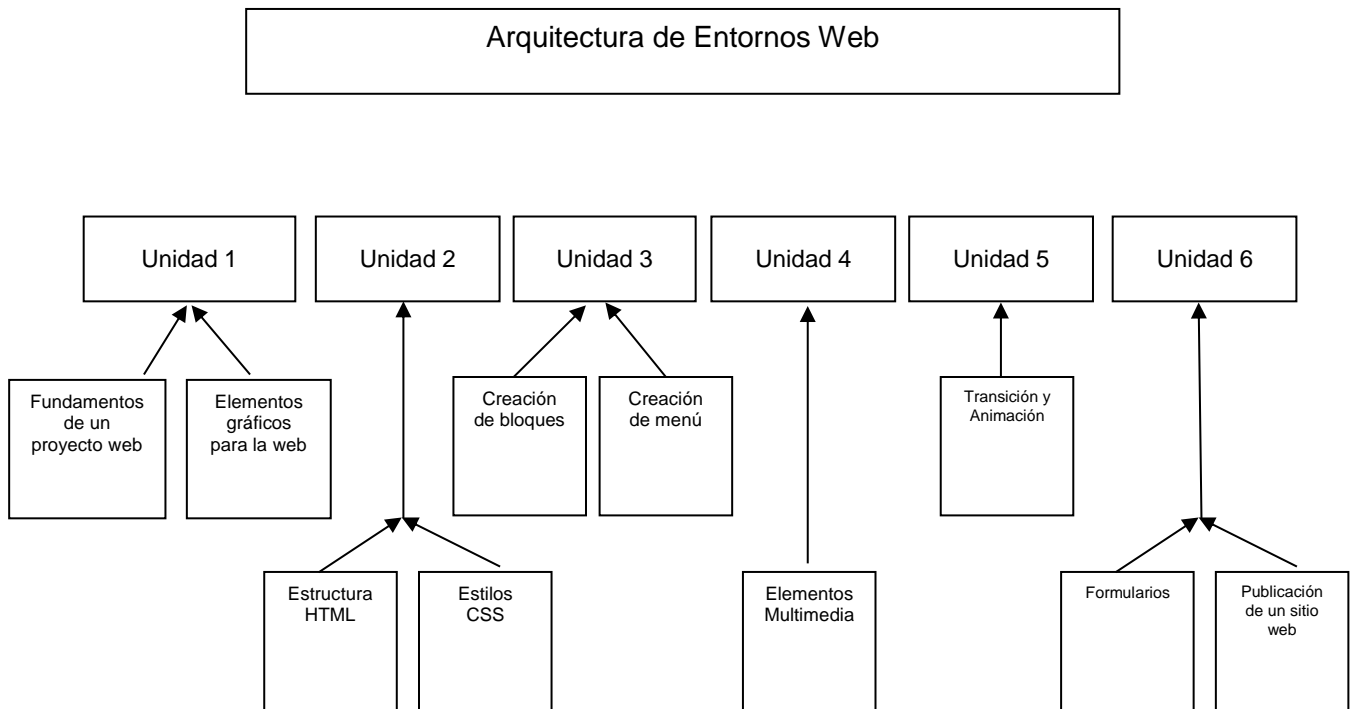
**Arquitectura de entornos Web** pertenece a la línea de tecnología y se dicta en las carreras de tecnología de la institución. El curso brinda un conjunto de herramientas de diseño y programación para la creación de páginas web.

El manual para este curso ha sido diseñado bajo la modalidad de unidades de aprendizaje, las que desarrollamos durante semanas determinadas. En cada una de ellas, hallará los logros que se deberá alcanzar al final de la unidad; el tema tratado, el cual será ampliamente desarrollado; y los contenidos, que debe desarrollar. Por último, encontrará las actividades y trabajos prácticos que deberá desarrollar en cada sesión, que le permitirán reforzar lo aprendido en la clase.

El curso es eminentemente práctico: consiste en el diseño de sitios web y programación de páginas web utilizando HTML5 y JavaScript. La primera parte de este manual explicamos acerca de los conceptos básicos de la Web y los fundamentos de un proyecto web. Luego, vamos a aprender a diseñar y construir páginas web para un sitio web aplicando las etiquetas HTML5, hojas de estilos CSS3, elementos de multimedia y formularios. A continuación aprenderemos a diseñar y programar páginas web utilizando el lenguaje de programación JavaScript; y por ultimo, aprenderemos a publicar un proyecto web en un sitio web.



# RED DE CONTENIDOS







# FUNDAMENTOS DE UN PROYECTO WEB Y ELEMENTOS GRAFICOS

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno entiende el concepto de un proyecto web y diseña los elementos gráficos necesarios para preparar un proyecto web.

## TEMARIO

### 1.1. Tema 1: Fundamentos de un proyecto web

#### 1.1.1 Generalidades

##### 1.1.1.1 Internet

##### 1.1.1.2 Servidores web

##### 1.1.1.3 Hosting y dominios

##### 1.1.1.4 Navegadores

#### 1.1.2 Etapas de un proyecto web

##### 1.1.2.1 Arquitectura de un proyecto web

##### 1.1.2.2 Mapa de sitio

##### 1.1.2.3 Wireframes.

### 1.2. Tema 2: Elementos gráficos para la web

#### 1.2.1 Manejo de un editor de imágenes

##### 1.2.1.1 Tipos de imagen, resolución, modos de color.

##### 1.2.1.2 Tamaño, recorte, selección, etc.

#### 1.2.2 Creación de banners, menús y botones.

## ACTIVIDADES PROPUESTAS

- Los alumnos desarrollan los laboratorios de esta semana
- Los alumnos desarrollan diseñan el mapa de sitio de su proyecto
- Los alumnos desarrollan diseñan su wireframe de baja y alta fidelidad.



## 1.1. FUNDAMENTOS DE UN PROYECTO WEB

### 1.1.1 GENERALIDADES

En la era de la globalización, la Internet se ha convertido en una herramienta indispensable para cualquier persona o profesional. Este medio se ha constituido como una de las principales vías de consulta de información, publicación y actualización de conocimientos y comunicación entre profesionales.

En esta era, las empresas, organismos estatales, publicaciones, programas de televisión o individuos no existen sino tienen su dirección de correo electrónico o su sitio web; la triple W (www) y la @ (arroba) ya forman parte de nuestras vidas, y se están haciendo cada vez más imprescindibles.

Internet y Web son dos términos que a menudo se utilizan indistintamente de forma que podemos caer en la confusión y mezcla de conceptos. Internet es un conjunto de redes de comunicación interconectadas entre sí, siguiendo el protocolo TCP/IP, que funcionan como una red única mundial y siendo la web un servicio más de Internet.

Acontinuación vamos a exponer los conceptos principales acerca de las generalidades de la web.

#### 1.1.1.1 Internet

Internet se le define como una red mundial de computadoras, distribuidos por el todo el mundo, las cuales se encuentran interconectadas a través de satélites, líneas telefónicas, fibra óptica, etc., donde utilizando un protocolo de comunicaciones común (TCP/IP), que permite a millones de usuarios estar conectados para compartir, intercambiar, publicar y extraer información.



Figura 1

Referencia: <http://definicion.de/internet/>

En la actualidad es un error considerar a Internet como si fuera sólo una red de computadoras. En este caso Internet es una gran fuente de información, donde podemos enviar mensajes, descargar programas, archivos y ficheros de texto; consultar catálogos de bibliotecas, solicitar libros, realizar compras.

Hay que tener en cuenta que todos los recursos que se pueden encontrar en Internet existen porque muchas personas en forma voluntaria han dedicado su tiempo en generarlos.

### **Protocolo de Internet**

El Protocolo de Internet es un conjunto de reglas formales que permite la transmisión de los datos entre dos computadoras. El ordenador conectado a una red usa protocolos para que envíe y reciba mensajes.



Figura 2

Referencia: [http://www.hipertexto.info/documentos/internet\\_tegn.htm](http://www.hipertexto.info/documentos/internet_tegn.htm)

El protocolo más utilizado es el TCP/IP (Protocolo de Control de Transmision TCP y Protocolo de Internet IP). Este protocolo especifica cómo las computadoras de puerta encaminan la información desde la computadora emisor hasta el computador receptor.



Figura 3

Referencia: <http://www.taringa.net/posts/info/12588433/Protocolos-de-Internet.html>

Actualmente existen varios protocolos de Internet entre los que explicamos los más utilizados:

**IPv6**, será la próxima generación de protocolos de Internet, este protocolo se ha desarrollado para ampliar la capacidad de



conexión debido al crecimiento de dispositivos y el aumento de equipos portátiles.

**FTP**, es un protocolo de transferencia de archivos de un ordenador a otro, de un servidor remoto a uno local y viceversa. Se precisa de un servidor de FTP y un cliente de FTP, estos pueden ser de acceso con una cuenta (login) o anónimo la cual acceder sin ninguna contraseña.

**HTTP**, protocolo de transferencia de Hipertextos, es el protocolo mas utilizado por los servidores de la World Wide Web, el cual permite el intercambio de información hipertextuales (enlaces) de las páginas web.

**SMTP**, protocolo de transmisión de correo simple, es el protocolo que nos permite recibir correos electrónicos y, junto con el protocolo POP (protocolo de oficinas de correo) es usado por los ordenadores personales para administrar el correo electrónico, nos permitirá bajarnos los mensajes a nuestra computadora.

**TELNET**, es un protocolo que permite la conexión remota a otro ordenador el cual nos permite manejarla como si estuviéramos físicamente frente al ordenador.

Otros conceptos:

**URLs**, localizador uniforme de recursos, representa la dirección electrónica para poder acceder a los recursos del servidor remoto. El tipo más común es el de las paginas web, con la dirección `http://`; también existen otras direcciones como `ftp://` para ubicar un recurso FTP para transferir archivos.

#### 1.1.1.2 Sitio Web

Se le conoce como **Site**, un sitio web es un conjunto de páginas web, relacionadas a un dominio de Internet el cual se puede visualizar a través de internet.



Figura 4

Referencia: <https://materialesmarketing.wordpress.com/2012/12/24/>

Generalmente, los sitios web poseen una estructura similar, lo cual nos puede servir para encontrar fácilmente algo que estamos buscando o para entender como funcionan los enlaces de las páginas.

En cada sitio web tiene una página de inicio (**Home**) que es el primer documento que ve el usuario cuando ingresa en el sitio web escribiendo su dirección URL en el navegador. Cada página web perteneciente al sitio web será visible o no (algunas pueden no ser visible al público en general), con el único objeto de publicar contenido.

La persona encargada de administrar un sitio web es el **WebMaster**. Esta persona es la responsable de todo el sitio web: decide las tecnologías que se van a utilizar, los servidores, la estructura de las páginas, el mapa de sitio. Es la única persona autorizada que decide si va o no la estructura o los contenidos del sitio.



Figura 5

Referencia: <http://blogs.zemos98.org/lacolinadeperalias/2017/01/18/>

Para ser un WebMaster no hay que estudiar ninguna carrera universitaria, son personas altamente capacitadas en el ámbito de la informática y la Internet, por lo cual conoce a fondo las tecnologías del mercado, trabaja en equipo junto a diseñadores, programadores, y otras personas para hacer alguna publicación.

Los sitios web se pueden clasificar en dos tipos:

**Sitios Estáticos:** Se denomina sitio estático a aquella que es básicamente informativa, donde el visitante y el administrador Web no pueden interactuar con la página para modificar su contenido. No acceden a una base de datos para obtener el contenido ni se requiere programación. Por lo general un sitio web estático es utilizado cuando el propietario del sitio no requiere realizar un continuo cambio en la información que contiene cada página.



Figura 6

Referencia: <http://www.taringa.net/posts/info/11539616/Diferencia-entra-una-web-estatica-y-una-web-dinamica.html>

**Sitios Dinámicos:** Son aquellos que pueden interactuar con el visitante y/o administrador Web, pudiéndose modificar el contenido de la página. Estos sitios acceden a una base de datos y requieren de programación Web: Java, ASP, ASP.NET, PHP, etc.



Figura 7

Referencia: <http://www.taringa.net/posts/info/11539616/Diferencia-entra-una-web-estatica-y-una-web-dinamica.html>

El administrador del sitio web podrá agregar, modificar y eliminar contenidos del sitio web a través de un “sistema web” con acceso restringido al público mediante usuario y contraseña, el cual se denomina BACK END.

Se asume a la hora de contratar el desarrollo de un sitio web, que el administrador especificará al desarrollador la cantidad de páginas discriminando si son dinámicas o estáticas que deberá tener el sitio web para su correcto.

### **Ventajas de un Sitio Web**

1. Alcance mundial de sus productos y servicios: a través del sitio Web Usted sus productos o servicios podrán ser conocidos por futuros clientes potenciales a nivel mundial. Las limitaciones geográficas ya no existen.

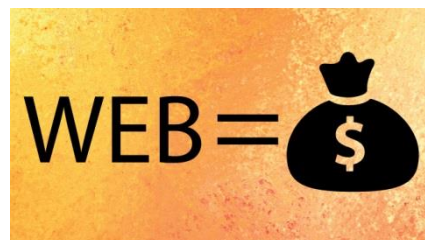


Figura 8

Referencia: <http://www.one-man-studio.com/ventajas-de-tener-una-pagina-web-como-herramienta-estrategica-de-venta-47-noticia>

2. Competir al lado de los más grandes: En la Internet su imagen puede ser tanto o más atractivo que cualquier empresa; por lo que podrá competir de igual a igual con las empresas más grandes.

3. Disponible las 24 horas, todos los días año: Un sitio web se encuentra disponible 24 horas del día, los 7 días de la semana, los 365 días del año, lo que permite que aún fuera del horario de trabajo, la empresa siga produciendo y creciendo.

4. Menos Costos Operativos: Se ahorra en gastos de publicidad, teléfono y personal, cualquier información que necesita el cliente la podrá consultar en su página Web.

5. Imagen Profesional de su empresa: Actualmente toda empresa debe tener presencia en la Web. No contar con una dirección en Internet, puede dar una imagen poco seria y profesional. Cada vez más el tener un sitio Web será definitivo para seguir en carrera.

### 1.1.1.3 Servidores Web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

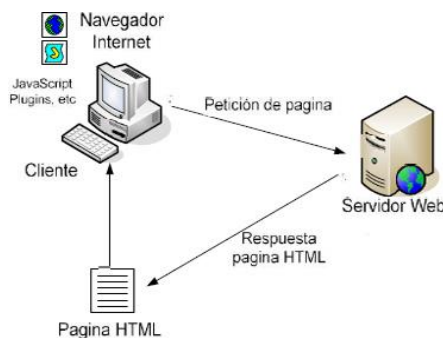


Figura 9

Referencia: <http://trimestrecuatrodeamarelysmaita.blogspot.com/2013/06/>

El funcionamiento de un servidor web básico basado en ejecutar infinitamente el siguiente bucle:

- Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
- Recibe una petición.
- Busca el recurso.
- Envía el recurso utilizando la misma conexión por la que recibió petición.
- Vuelve al segundo punto.

Un servidor web que siga el esquema anterior cumplirá todos los requisitos básicos de los servidores HTTP, aunque sólo podrá servir ficheros estáticos.

En este punto es necesario aclarar lo siguiente: mientras que comúnmente se utiliza la palabra **servidor** es para referirnos a una computadora con un software servidor instalado, el cual se

encargar de “alojar” los sitios web que visitamos en la Internet y son estas computadoras las que proporcionan al usuario la interacción con el sitio en cuestión.

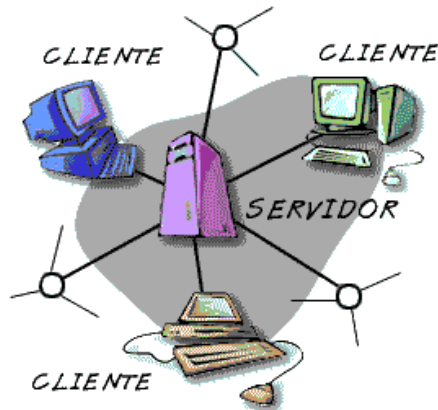


Figura 10

Referencia: <http://valbuena.wikispaces.com/FUNCIONAMIENTO+DE+UN+SERVIDOR+WEB>

### **Porqué son necesarios los servidores**

Los servidores son como la columna vertebral de la estructura de Internet. La industria del Web hosting es simplemente la forma de alquilar esos espacios de memoria y administración de datos.

Sin este sistema de leasing, la Internet sería sólo una pequeña parte de lo que es hoy por hoy, ya que sería imposible para la mayoría de los dueños de sitios Web comprar un servidor para uso propio.

### **Tipos de servidores**

Servidor Dedicado: que trata de una computadora dedicada exclusivamente al sitio del cliente, para aplicaciones de alta demanda. Entre las características tenemos:

- Son de pago superior a fin de tener un servidor para sí.
- Mayor capacidad en el disco duro, alta tasa de transferencia
- Son más difícil de manejar

Servidor Conjuntamente: se trata de tipo de servidor similar al dedicado. La diferencia es que usted es dueño del servidor, es decir, lo compra en lugar de alquilarlo de una empresa de servicio hosting. Entre las características tenemos:

- No son gratuito se paga una tarifa por el ancho de banda y tasas de mantenimiento
- Se beneficia por contar con una alta tasa de velocidad.
- El servidor se encuentra ubicado físicamente en un centro de datos

Servidor Compartido: se trata de un ordenador dentro del cual aloja más de un sitio web. Entre las características tenemos:

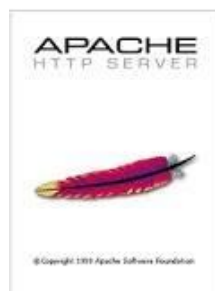
- La mayoría de los servidores de este tipo son gratuitos
- Todos los sitios web deben compartir recursos de una misma computadora, de manera desigual
- Se controlan fácilmente
- Uno de los inconvenientes es que algunos sitios web requieren de más recursos por ser de más rendimientos que las otras

Servidor Virtual Privado (VPS): se trata de un ordenador el cual divide todos sus recursos entre los sitios web que aloja. Entre las características tenemos:

- No son gratuito y su precio no es muy caro
- Su manejo es sencillo, te lo dan casi todo hecho
- Si el procesador se sobrecarga, fallarán los sitios web
- Se comparte el HDD y la RAM pero no el procesador

### **Tipos de servidores Web**

Apache: Este es el más común y mas utilizado en todo el mundo. Además, es gratuito y de código abierto, lo que implica que corre en cualquier plataforma.



Microsoft IIS: Solamente funciona sobre la plataforma Windows. Para que trabaje sobre otro sistema tendrás que utilizar una máquina virtual.



Nginx: es un servidor web ligero y corre sobre un sistema Unix y Windows.

Lighttpd: Está especialmente pensado para hacer cargas pesadas sin perder balance, utilizando poca memoria RAM y poca de CPU.

Sun Java System Web Server: Es multiplataforma y se distribuye con licencias de código abierto.

#### 1.1.1.4 Hosting y Dominios

##### Hosting

El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

Esta palabra es una analogía de “hospedaje o alojamiento en hoteles o habitaciones” donde uno ocupa un lugar específico, en este caso la analogía alojamiento de páginas web, se refiere al lugar que ocupa un sitio web, sistema, correo electrónico, archivos etc. en un servidor que por lo general hospeda varias aplicaciones o páginas web.



Figura 11

Referencia: <http://chiapashosting.com/hosting-o-alojamiento-web.html>


##### Principales Ventajas de un Hosting

- Tener un servicio de hosting disminuye los costos a los usuarios, puesto que un grupo de usuarios (los cuales acceden al mismo servicio) se encargará de compartir los gastos del servidor, mantenimiento y sus actualizaciones
- Algunos de los servidores de web hosting no cuentan con un administrador determinado, hecho que también se encarga de disminuir los costos de manera significativa; también existen otros servidores de web hosting que cuentan con equipos de profesionales y con personal de servicio al cliente, especialmente dedicados a responder las necesidades más específicas de los clientes.
- Los servicios de web hosting son indispensables para todo tipo de empresas que surgen en la web, no importa si son servicios de web hosting gratuitos, dedicados o compartidos, lo importante es acceder al hosting adecuado para nuestras necesidades.

## Qué debo tener en cuenta antes de contratar un servicio hosting

Como en la mayoría de los distintos servicios, todo depende de lo que precises. Hay servicios de hosting muy económicos, que no poseen ningún tipo de garantía sobre tus datos, ni te aseguran una estabilidad mínima de tu servicio; y también hay otros servicios que incluyen seguramente aplicaciones y servicios que no necesitamos.

Por supuesto, los servidores de pago prestan un mejor servicio, más personalizado y ofrecen soporte técnico, paneles de control amigables, así como plantillas y demás comodidades completamente instaladas en nuestro servicio de alojamiento.



Emprendedor Junior	Emprendedor Senior	Profesional	Empresarial
<b>\$6.99</b> Por mes	<b>\$9.99</b> Por mes	<b>\$14.99</b> Por mes	<b>\$19.99</b> Por mes
✓ 1 hora de servicio web personalizado	✓ 1 hora de servicio web personalizado	✓ 2 horas de servicio web personalizado	✓ 3 horas de servicio web personalizado
✓ 2GB de almacenamiento super rápidos	✓ 4GB de almacenamiento super rápidos	✓ 6GB de almacenamiento super rápidos	✓ 8GB de almacenamiento super rápidos
✓ 40GB de Transferencia Mensual	✓ 60GB de Transferencia Mensual	✓ 120GB de Transferencia Mensual	✓ 200GB de Transferencia Mensual
✉ 10 Cuentas de correo en la nube (recomendado) o 10 Cuentas de correo POP3	✉ 10 Cuentas de correo en la nube (recomendado) o 25 Cuentas de correo POP3	✉ 10 Cuentas de correo en la nube (recomendado) o 50 Cuentas de correo POP3	✉ 10 Cuentas de correo en la nube (recomendado) o 100 Cuentas de correo POP3
✗ Migración cPanel / Plesk	✓ Migración cPanel	✓ Migración cPanel / Plesk o cualquier gestor	✓ Migración cPanel / Plesk o cualquier gestor
✗ Dirección IP dedicada	✗ Dirección IP dedicada	✓ Dirección IP dedicada	✓ Dirección IP dedicada
✗ CDN Brasil, USA, Europa	✗ CDN Brasil, USA, Europa	✗ CDN Brasil, USA, Europa	✓ CDN Brasil, USA, Europa
✗ Certificado SSL	✗ Certificado SSL	✗ Certificado SSL	✓ Certificado SSL
<a href="#">Contratar</a>	<a href="#">Contratar</a>	<a href="#">Contratar</a>	<a href="#">Contratar</a>

**Figura 12**  
Referencia: <https://hosting-peru.pe/alojamiento-web/>

Para contratar un servicio hay que tener en cuenta una gran cantidad de factores, como:

- La cantidad de espacio en MB que ofrece el servicio para alojar nuestras páginas web
- La capacidad de gestionar los archivos vía FTP
- Soporte para scripts perl y python, php, ASP, etc
- Tipo de dirección web que ofrecen
- Tasa de transferencia de archivos mensuales



## Dominio

Un dominio de Internet es un nombre de un sitio web. Cada dominio tiene que ser único en Internet, por ejemplo “elcomercio.pe” es el nombre del dominio de la página web del diario El Comercio.



Figura 13

Referencia: <http://blog.ozongo.com/cuantas-extensiones-necesito-para-el-dominio-de-mi-empresa/>

Todos los servidores y páginas de Internet tienen una dirección numérica que se conoce como dirección IP (Protocolo de Internet), por ejemplo 50.17.245.112

Los dominios fueron creados para evitar el que tuviéramos que recordar las direcciones numéricas de las páginas y servidores web. De forma que cuando escribimos en internet el dominio elcomercio.pe, el servidor de DNS (Servidor de Nombres de Dominio) del proveedor de web hosting del dominio elcomercio.pe nos proporciona la dirección IP 50.17.245.112 y nuestro navegador se va directamente a esa dirección numérica.

### Elementos de un dominio

Un dominio de Internet está compuesto por 2 elementos:

- Nombre del dominio
- Extension o terminación del dominio

El nombre hace referencia a la marca o una palabra resumen que indique el tipo de actividad o información que brinda la web, por ejemplo [www.pepsi.com.ar](http://www.pepsi.com.ar). Esta dirección hace referencia al nombre de una marca mundial de refrescos.

La terminación o extensión de los dominios, indica la ubicación geográfica donde se ubica el sitio web así como el tipo de organización o fin que tiene. Es aquí donde podemos distinguir dos tipos de extensiones:



Figura 14

Referencia: <http://www.quees.info/que-es-un-dominio.html>

**Territoriales:** indican a que país o región pertenece la página, actualmente existen un total de 243 países y está compuesto por dos letras que corresponde a la abreviación del nombre del país:

- .es, España
- .pe, Peru
- .us, Estados Unidos

**Internacionales:** hacen referencia al tipo de actividad que ofrece el sitio web. A diferencia de los territoriales, éstos están compuestos por más de dos letras.

Dominio	Descripción
.biz	Prevista para ser usado por negocios
.com	Son los dominios más extendidos en el mundo. Sirven para cualquier tipo de página web, temática
.edu	para servicios de Educación
.gov y .gob	para gobierno y entidades públicas
.info	para información
.mil	para el Departamento de Defensa de los Estados Unidos
.name	para nombres de personas
.net	para infraestructura de red e Internet
.org	para organizaciones
.tel	para servicios de comunicación por internet
.travel	para páginas de la industria de viajes y turismo
.aero	Sector aeronáutico
.edu	Universidades, educación
.jobs	Empleo

### 1.1.1.5 Navegadores

Un navegador o navegador web (del inglés, web browser) es un programa que permite visualizar la información que contiene una página web la cual puede estar alojada en un servidor dentro de la World Wide Web o en uno local.

El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.



Figura 15

Referencia:

<http://www.adrformacion.com/cursos/dreamcs3/leccion1/tutorial3.html>

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, reproducir recursos multimedia incrustados y ejecutar acciones. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web).

### Navegadores más utilizados



Figura 16

Referencia: <http://quelaptopcomprar.blogspot.com/>

Google chrome: Creado por la empresa Google. Se lo puede obtener de forma gratuita, y se caracteriza por tener una interfaz de usuario que resulta eficiente y de gran sencillez. Este navegador es valorado por sus usuarios debido a la seguridad, claridad, sencillez, rapidez y estabilidad que proporciona. Además, le otorga a los usuarios privacidad en sus navegaciones al borrar los archivos cookies y no registrar las actividades.

Internet Explorer: Integrado al sistema operativo Windows, lo que trae como consecuencia que no pueda ser borrado por el usuario. A diferencia del Google chrome, este navegador no es considerado demasiado seguro, aunque presenta avanzadas opciones en cuanto a privacidad y seguridad.

Mozilla Firefox: Incluye, corrector ortográfico, pestañas, sistema de búsqueda integrado que usa el motor de búsqueda que quiera el usuario. Se caracteriza por su estabilidad, sencillez, seguridad y rapidez, y permite al usuario modificar su apariencia y estilo.

Opera Browser: Posee soporte estándar, gran velocidad y es de fácil uso. Una de las grandes ventajas que posee es que se lo actualiza y mejora constantemente. También contiene un buscador y pestañas. Una ventaja que tiene es que eliminó la presencia de avisos publicitarios. Opera Browser está integrado a las consolas Nintendo y a teléfonos celulares.

Safari: Fue creado por Apple. Una de las ventajas que presenta es la posibilidad de enviar por mail páginas web y almacenarlas a pesar de que estas hayan sido eliminadas de la red. Se caracteriza por ofrecer controles parentales muy eficientes, proporcionar seguridad y proteger los datos.

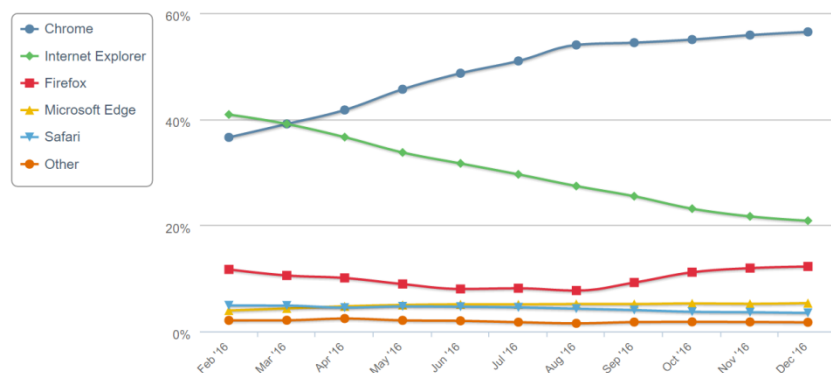


Figura 17

Referencia: <https://www.genbeta.com/a-fondo/asi-esta-el-ranking-de-los-navegadores-y-sistemas-operativos-mas-utilizados-al-empezar-2017>

### 1.1.2 ETAPAS DE UN PROYECTO WEB

Desarrollar un sitio web efectivo requiere una planeación bien elaborada. Puede ahorrar tiempo y dinero durante el proceso del proyecto si logra entenderlo en su totalidad antes de comenzar.

A pesar de que los sitios varían en rango de contenido, desde unas cuantas páginas hasta sitios mucho más complejos con características sofisticadas, el proceso de desarrollo se mantiene igual.

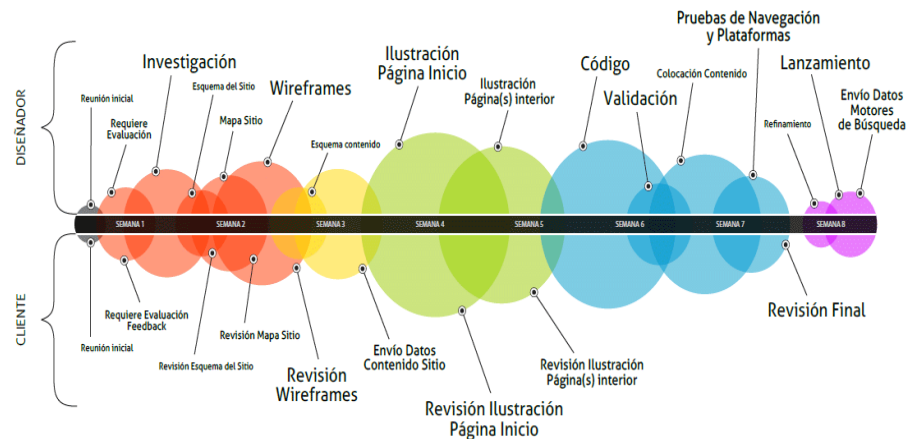


Figura 18

Referencia: <http://www.n4gash.com/2016/infografa-pasos-diseno-pagina-web/>

Las etapas de un proyecto de diseño de sitios Web son:

1. **Estrategia:** Es la primera etapa a definir antes de tomar cualquier acción. Se necesita decidir cuál es el propósito del sitio web, cuáles son las necesidades del negocio que motivan el proyecto. Es importante definir:
  - Objetivos y metas.
  - Analizar y recoger la información
  - Lista de puntos para producción (tecnología, servidor web, presupuesto)
  - Designar un editor del sitio
2. **BenchMarking:** conocer qué está haciendo la competencia y como está posicionado su negocio en la Internet. Se analizan sitios web relacionados con el área del proyecto, determinando elementos y funcionalidades destacados que puedan ser utilizados para ofrecer un sitio web más competitivo.



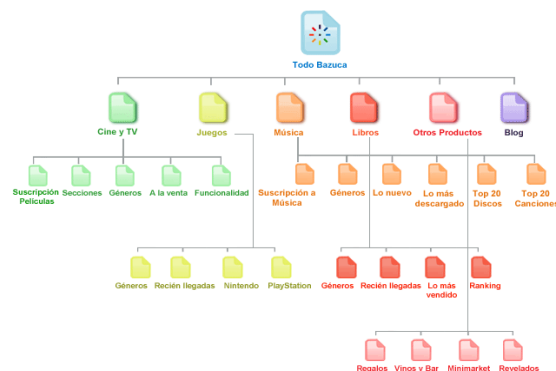
Figura 19

Referencia: <http://www.addwebproactive.com/>

Es recomendable enfocarse en aquellas que tengan un mayor tamaño e imagen de marca que la de nuestro cliente, incluso a los líderes del mercado.

3. **Arquitectura de la información:** Con toda la información que se va a contar de las reuniones realizadas con los clientes, creamos un mapa del sitio (**SiteMap**) para el proyecto donde mostramos cómo será la estructura de secciones de su nueva web. Este proceso ayuda a comprender qué y cómo vamos a comunicar y establecer las funcionalidades que los usuarios necesitarán.

Los contenidos bien planificados y bien estructurados nos ayudarán a crear tanto el sistema de interacción como el diseño visual.



**Figura 20**

Referencia: <http://informaticayserviciosdeinternet.blogspot.com/2012/05/mapa-de-sitio.html>

4. **Diseño de prototipos:** A partir de la priorización de contenidos, trabajamos con el cliente a través de un proceso iterativo para diseñar un prototipado o boceto muy esquemático (en inglés, wireframe) de las principales páginas de la web.



**Figura 21**

Referencia: <http://teamddm.com/articles/wireframes-getting-a-shared-understanding>

Esto ayudará al cliente a conocer cómo se organizará y será mostrado el contenido en dispositivos móviles, tablets y ordenadores de sobremesa.

5. **Construcción:** En este punto comenzamos a crear la plantilla de la web con HTML5.



Figura 22

Referencia: <http://laminaliagalicia.es/?p=1>

Este paso es a menudo un trabajo de colaboración estrecha entre diseñadores y desarrolladores. La codificación adaptativa empezará con el diseño web y usaremos la lista de priorización de contenidos o el prototipado como mapa de ruta para priorizar los contenidos. Incorporaremos los diferentes estilos de diseño según lo especificado en la guía de estilos mediante CSS3 y nos aseguraremos de que cumple con los estándares de W3C. Usaremos JQuery y JavaScript para eventos interactivos, en lugar de Flash o Silverlight. También aplicaremos las mejores prácticas para facilitar la lectura online y la impresión de contenidos añadiendo el código específico en CSS.

6. **Pruebas de usabilidad y refinamiento:** Las pruebas de usabilidad son evaluaciones que se realizan a sitios web con el objeto de determinar si son intuitivos y fáciles de navegar. Estas pruebas se realizan con usuarios reales y con usuarios expertos.



Figura 23

Referencia: <http://www.condo-consulting.com/test-de-usabilidad-que-son-y-para-que-me-sirven.b.aspx>

Realizar pruebas de usabilidad a su página web permitirá saber si la información está bien estructurada y si generará una experiencia positiva al usuario en cada visita. Al realizar esta prueba se evalúa la navegabilidad, arquitectura de la información, entradas y datos, formularios, credibilidad y confianza de su página web.



7. Publicación en la Web: Este será el último paso, subir el proyecto a la Internet, para ello debemos tener en cuenta lo siguiente:
  - Contratar un hosting, la empresa dispone de distintas alternativas: alojamiento virtual, tener servidor propio, servidor dedicado.
  - Si quiere genera tráfico hacia el sitio web, la empresa debe iniciar una labor de promoción del sitio.
  - Para que el sitio se visitado en forma continua, la empresa deberá realizar continuas actualizaciones y revisiones de sus paginas.



Figura 24

Referencia: <http://www.callegranvia.com/empresariales/info/promover-su-negocio-con-la-publicacion-de-articulos/>

### 1.1.2.1 Arquitectura de un proyecto Web

Se utiliza el término arquitectura web, para definir una tarea que requiere conocimientos técnicos de construcción, funcionales y de diseño para sitios o páginas web. Con base en las preferencias del contenido se crea un mapa de la arquitectura del sitio para demostrar visualmente cómo se organizan el contenido y la estructura del sitio.

El objetivo de esta fase es poder generar un mapa detallado de las secciones que se deberán construir y los elementos con los que estas deberían contar. Es muy importante poder incluir en estos esquemas información del tipo de dato a trabajar y la jerarquía del contenido. una vez que se generen los esquemas que deberán ser entregados al proceso de diseño.

### 1.1.2.2 Mapa de Sitio

Un Sitemap o Mapa del Sitio es la estructura de enlaces de páginas web que definirá la buena o mala navegación de un sitio web.

El mapa de un sitio web puede ser un documento que se utiliza para planificar el diseño de la web, o puede ser una página web donde se listan todas -o las más importantes- páginas web de un sitio (generalmente organizadas de alguna manera).



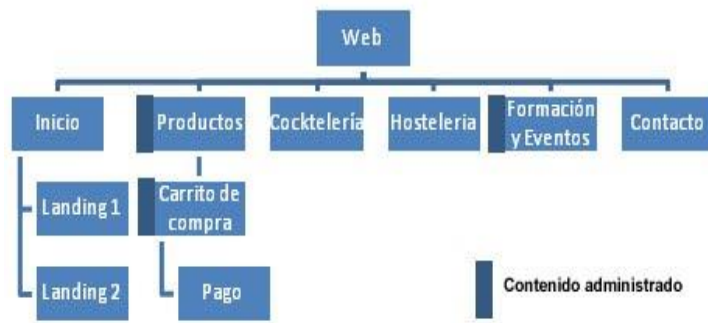


Figura 25

Referencia: <http://marketinges.es/tag/sitemap/>

Empezando por la página de Inicio hasta la página de contacto, todo sitio web debe tener un mapa para que la navegación del usuario web sea lo más fácil posible.



Figura 26

Referencia: <http://es.slideshare.net/christian.bonner/arquitectura-de-la-informacin-de-sitios-web>

### Importancia

Un mapa de sitio es importante fundamentalmente para ayudar a los buscadores a que encuentren nuestra web y a que indexen sus páginas más rápidamente.

Es como si a un buscador le dieramos un índice de nuestro libro (nuestro libro es nuestra web) y le ahorráramos el esfuerzo de tener que buscar él mismo donde está cada cosa.

### Características

- Consistentes: el sistema debe ser similar en todo el sitio, en lo referido a su ubicación y disposición en las páginas. Esto se aplica también a aquellas instituciones que pueden tener más de un sitio web.
- Uniforme: el sistema debe utilizar similares términos con el fin de que el usuario que lo vea en las páginas, confíe en que sus opciones llevan siempre hacia los mismos lugares dentro del sitio.
- Visible: debe ser una guía permanente en el área en que se encuentre el usuario dentro del sitio.

### Ventajas

- Resumen de la información y accesibilidad en un enlace directo
- Facilidad de navegación
- El usuario permanece mas en la Web
- La navegación resulta una experiencia agradable
- Una estructura clara y sencilla

### Técnica para construir un sitio web

Una de las técnicas más populares utilizadas por los arquitectos de información es la conocida card sortung (tarjetas ordenadas).



**Figura 27**

Referencia: <http://es.slideshare.net/christian.bonner/arquitectura-de-la-informacin-de-sitios-web>

Esta técnica se basa en la observación de como los usuarios agrupan y asocian entre sí un numero predeterminado de tarjetas etiquetadas con las diferentes categorías del sitio web.

De esta forma, partiendo del comportamiento de los propios usuarios, es posible organizar y clasificar la información de un sitio web conforme a su modelo mental.

### **1.1.2.3 Wireframe**

Como desarrolladores o diseñadores web, cuando nos enfrentamos a un nuevo proyecto con un cliente es de vital importancia lograr una buena comunicación con el cliente y detectar realmente sus necesidades; pero esto es un problema cuando en algunos casos el cliente no sabe lo que necesita realmente o no puede explicarlo de manera adecuada.

Para evitar el tener que volver atrás en nuestros proyectos y para mostrarle al cliente una mejor visualización del proyecto, nace la idea de los mockups, maquetas o wireframes.

Un wireframe básicamente es un boceto básico y de baja calidad del desarrollo de una página web o el diseño de una interfaz, cuya finalidad es el mostrar al cliente un boceto rápido para facilitar la comunicación entre cliente y desarrollador.

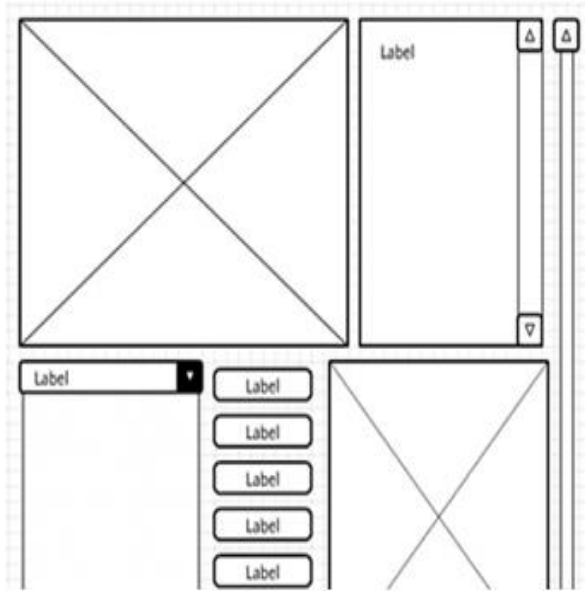


Figura 28

Referencia: <http://tednologia.com/que-es-un-mockup-o-wireframe/>

El wireframe tiene como objetivo proporcionar la visualización del diseño o las funcionalidades de un site de manera netamente referencial y poder definir objetivos en una etapa temprana del proyecto, incluso hacer modificaciones de manera rápida, para evitar hacer estos cambios sobre un site o diseño real.

A ellos debe sumarse, para ciertos proyectos en que tan relevante como ubicar los elementos en pantalla es simular el funcionamiento, los wireframes funcionales, que no se limitan a una estructura estática, sino que también permiten la navegación.

Existe una variedad de software que nos permite crear estos tipos de wireframes:

Nombre	Ubicación
Balsamig	<a href="http://www.balsamig.com/products/mockups">http://www.balsamig.com/products/mockups</a>
Mockingbird	<a href="https://gomockingbird.com/">https://gomockingbird.com/</a>
Lovely Charts	<a href="http://lovelycharts.com/">http://lovelycharts.com/</a>
Mockflow	<a href="http://www.mockflow.com/">http://www.mockflow.com/</a>
Gliffy	<a href="http://www.gliffy.com/uses/wireframe-software/">http://www.gliffy.com/uses/wireframe-software/</a>

### Tipos de Wireframe

Podemos tener distintos tipos de prototipos, que en ningún caso incluyen el diseño gráfico puesto que este se aplica en una etapa posterior.

1. **Wireframe de baja calidad:** son aquellos prototipos que representan cómo estarán organizados los elementos en las páginas del sistema sin entrar en detalle.
  - a. **Sketching:** un sketch es un boceto rápido e informal que se realiza en papel para transmitir una idea o concepto con rapidez y claridad. Suele realizarse durante las entrevistas iniciales como técnica eficaz para comunicar al cliente conceptos o proponer alternativas a un problema; también es muy útil durante un brainstorming o en las reuniones con el equipo de trabajo. Es una forma eficaz de comenzar el prototipado de un sitio web, pues permite trabajar ágilmente con varias ideas y esquematizar las páginas.

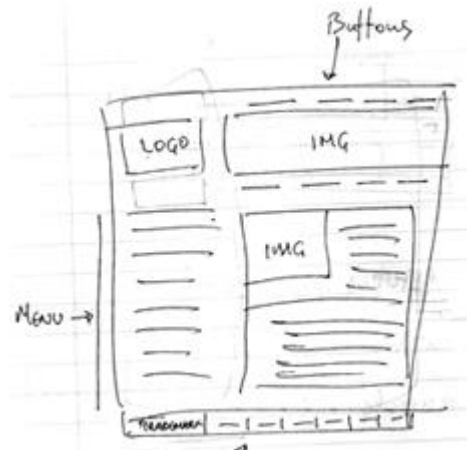


Figura 29

Referencia: <http://tednologia.com/que-es-un-mockup-o-wireframe/>

- b. **Wireframe:** permite crear una referencia visual de la estructura de una página web, definiendo al detalle el contenido y su distribución visual, organizando así la información a nivel de página.

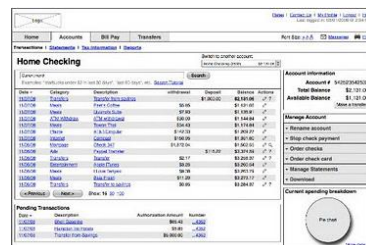


Figura 30

Referencia:

<http://olgacarreras.blogspot.com.es/2007/02/wireframes.html>

2. **Wireframe de alta fidelidad:** son prototipos o maquetas dinámicas, normalmente en (X)HTML, que simulan o tienen implementadas partes del sistema final a desarrollar. Pueden ser de muy distintos tipos, completo o parcial, horizontal o vertical, etc.

#### Ventajas de crear Wireframes

- Comunicación entre todos los componentes del equipo de desarrollo y los usuarios
- Participación, para integrar activamente a los usuarios en el desarrollo.
- Dan soporte a los diseñadores a la hora de escoger entre varias alternativas.
- Permiten a los diseñadores explorar diversos conceptos del diseño antes de establecer los definitivos.
- Permiten evaluar el sistema desde las primeras fases del desarrollo (facilitan la exploración de ideas sobre nuevos conceptos tecnológicos).
- Son esenciales para la documentación, tanto de conceptos funcionales del sistema como de tareas concretas del mismo.
- Son el primer paso para que ideas abstractas sean concretas, visibles y testables.
- Fomentan la iteratividad.
- Mejoran la calidad y la completitud de las especificaciones funcionales del sistema.
- Son herramientas de propósito general, pues sirven para comprobar la fiabilidad técnica de una idea, clarificar requisitos que quedaron “indeterminados” o ver como responde con el resto de la aplicación.

## LABORATORIO 1.1

### Creando un mapa de sitio con ADOBE MUSE

Se pide diseñar un mapa de sitio utilizando el aplicativo Adobe MUSE.

#### Pasos a seguir:

Para crear un sitio Web, inicie el aplicativo Adobe MUSE

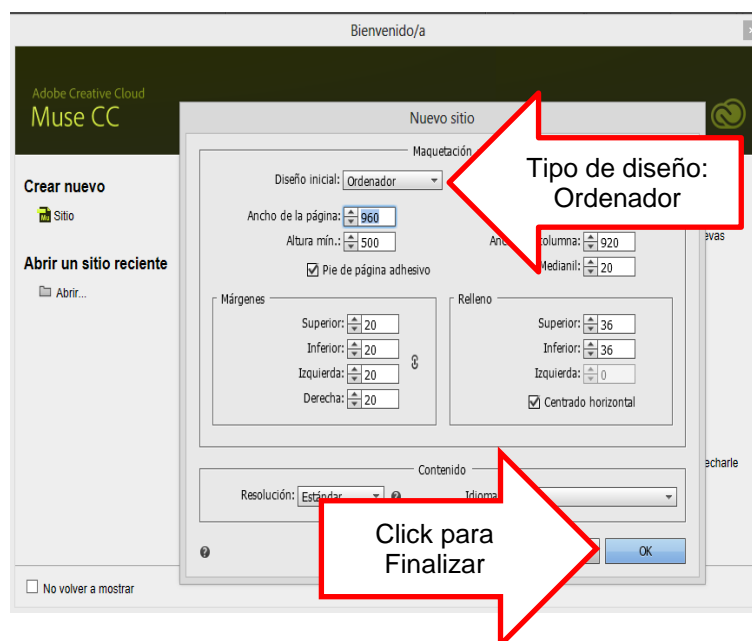


Para crear un nuevo sitio, hacer click en la opción **Crear nuevo Sitio**, tal como se muestra.



Configura el nuevo sitio, selecciona el diseño Inicial del sitio: Ordenador, tal como se muestra.

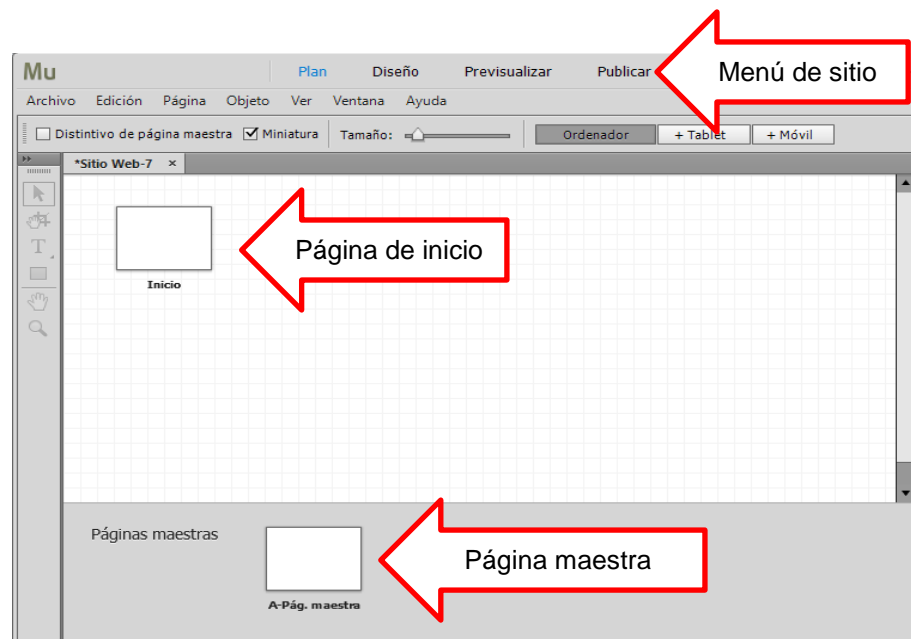
Al finalizar selecciona la opción OK.



## IDE del aplicativo

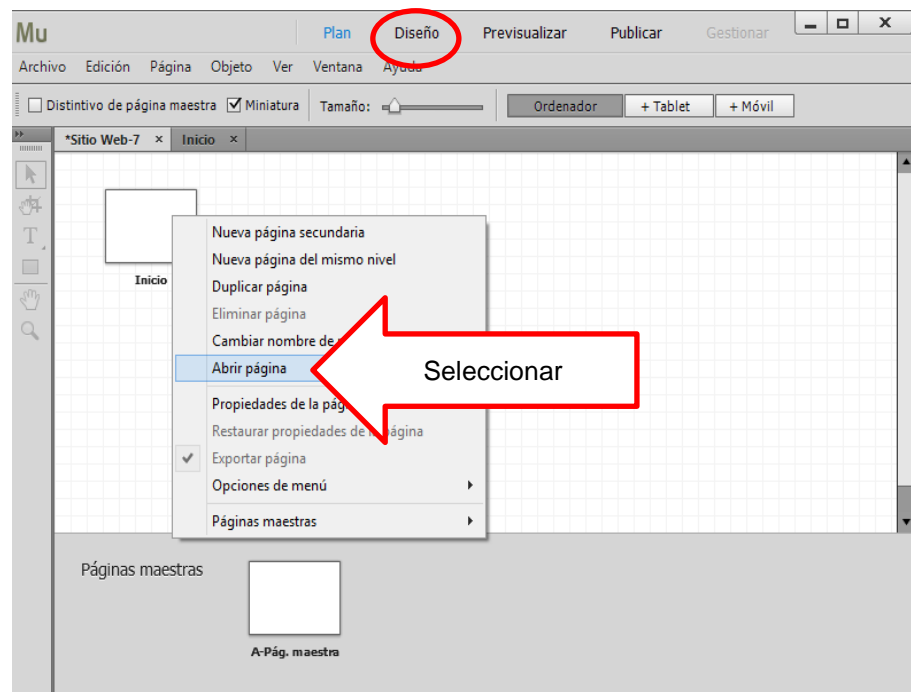
Al crearse el sitio, se muestra la siguiente ventana, tal como se muestra.

Para visualizar el diseño de las páginas, selecciona la opción **Miniatura**.



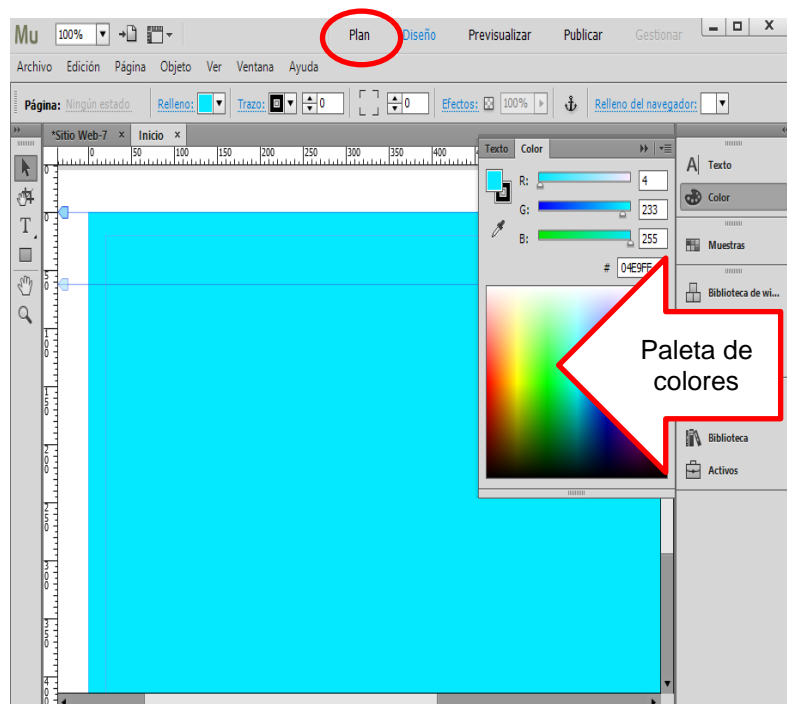
## Diseñando una página

Para diseñar una página, desde el menú de contexto seleccionar la opción Abrir página, o bien hacer click en la opción **Diseño**, tal como se muestra.



Para asignar color a la página, seleccione desde la paleta de colores, donde al seleccionar se muestra en el diseño de la página, tal como se muestra.

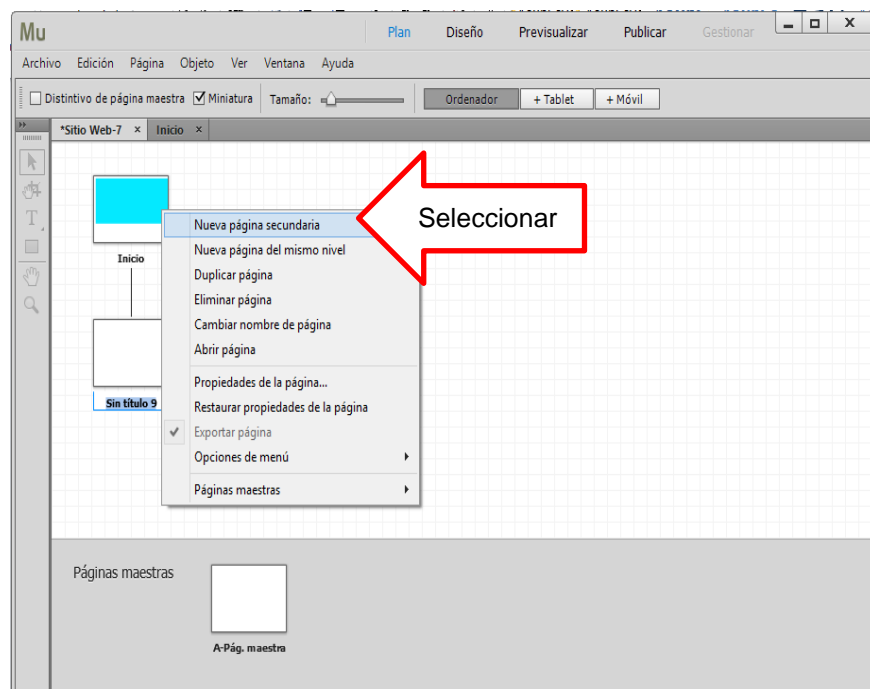
Para regresar al sitio Web, presiona la opción **Plan**.



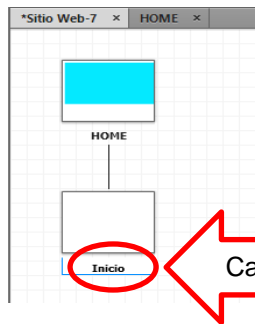
## Agregando páginas a siguiente nivel.

Para agregar páginas en el siguiente nivel (páginas secundarias) desde el menú contextual selecciona la opción Nueva página secundaria tal como se muestra.

Asigne el nombre a la página haciendo doble click al nombre.





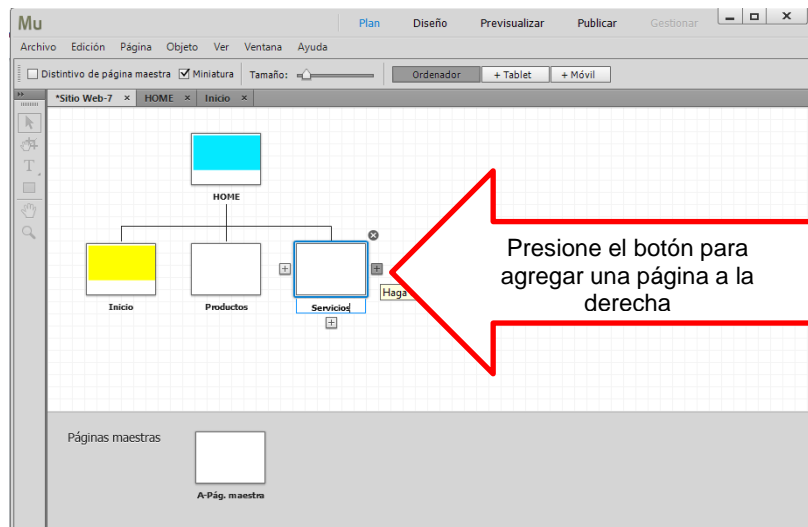


Cambie el nombre de la página y diseñe su contenido

Cambie el nombre

## Agregando páginas al mismo nivel

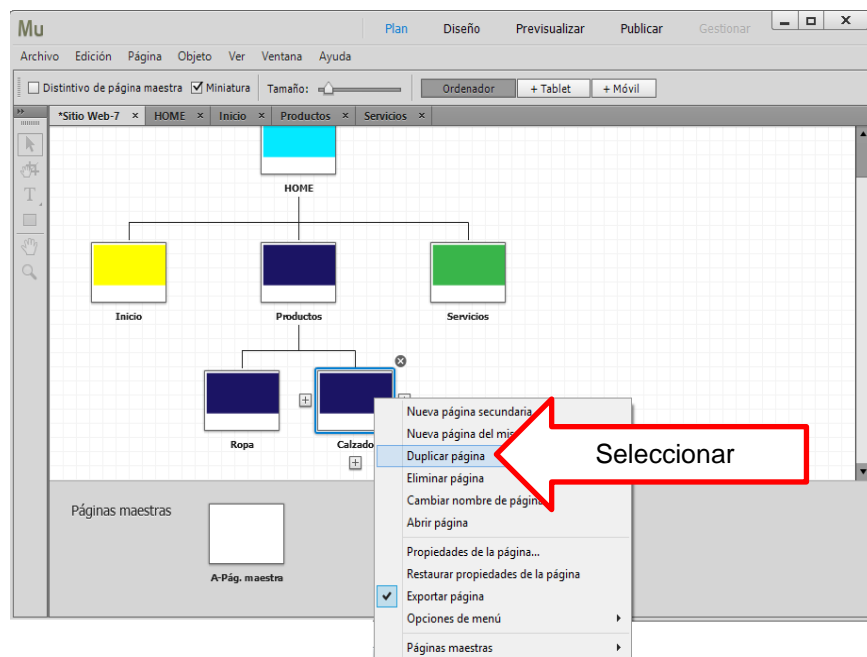
Para agregar páginas al mismo nivel, presione el botón (mas) del lado derecho de la página y se agrega una página en el mismo nivel, tal como se muestra.



Presione el botón para agregar una página a la derecha

## Duplicando páginas

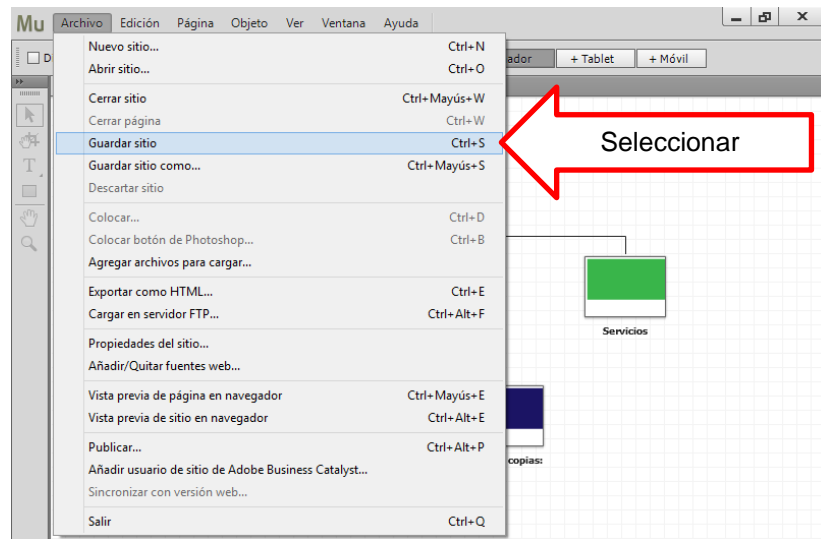
Para crear páginas con el mismo diseño, debemos “duplicarlas”; para ello seleccione una página y seleccione, desde el menú contextual, la opción Duplicar página, tal como se muestra



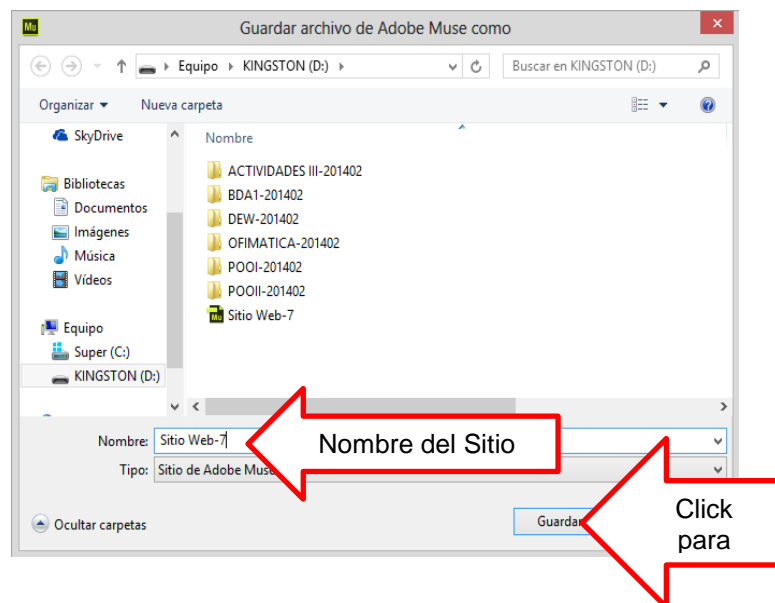
Seleccionar

## Grabando el sitio

Para guardar el sitio, seleccione desde la opción ARCHIVO, la opción Guardar Sitio



Selecciona la carpeta donde se guardará el sitio y presione el botón GUARDAR



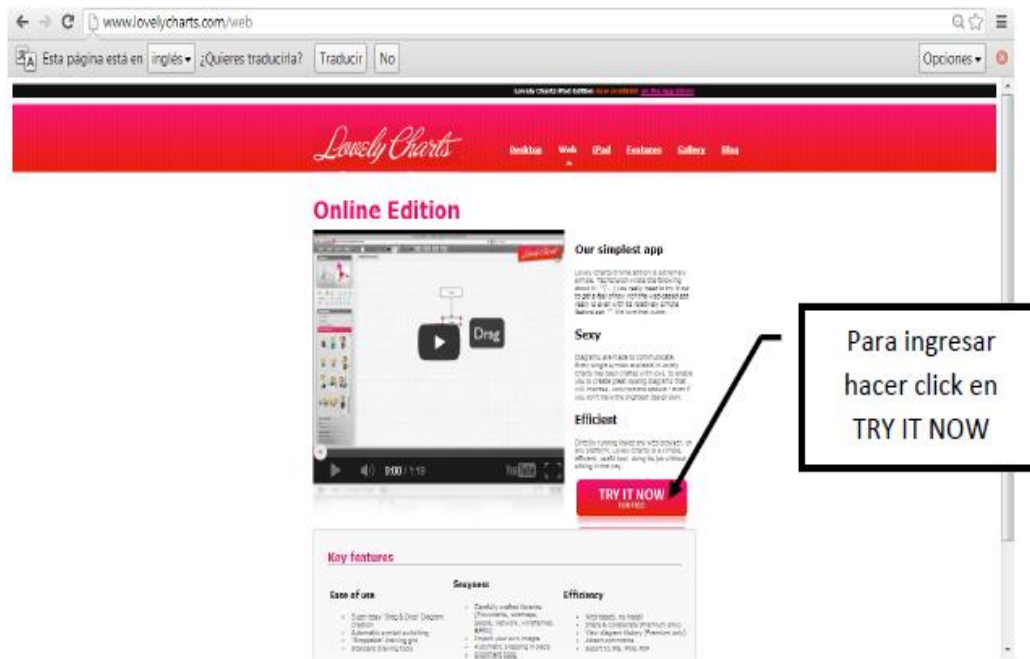
## LABORATORIO 1.2

### Creando un mapa de sitio con LOVELY CHARTS

Se pide diseñar un mapa de sitio utilizando el aplicativo Lovely Charts.

## ACCESO A LA PLATAFORMA

Ingresa a la plataforma. Para ingresar al LOVELY CHARTS, escriba la siguiente dirección URL <http://www.lovelycharts.com/web>



Si no tienes una cuenta, presiona el botón **Try It Now**, para registrarte.



Registras tus datos para ingresar al aplicativo, tal como se muestra.

**1. Register**

Hi there! Signup to begin using Lovely Charts in a minute, for free!  
All we need is your (valid) email address and a password to identify you, and your name to greet you properly :)

Name: \* Juan

E-mail: \* juan.cabrera@gmail.com

Password: \* \*\*\*\*

Confirm Password: \* \*\*\*\*

☐ Subscribe to our Lovely newsletter!

\* ☒ I agree to the [Terms of service](#)

**Annotations:**

- Ingresas los datos solicitados (points to the input fields)
- Selecciona la opción I agree (points to the checkbox)
- Presiona el botón Register (points to the Register button)

Una vez que te has registrado, la plataforma te envía un código por tu correo. Ingresa el código que le envía el sitio, e ingresarás a la aplicación web. Selecciona SiteMaps de la opción Create New.

**Welcome!**

Hello damasol!

**First time here?**

Apparently you haven't saved any diagram yet, so maybe this is your first time here. We've put a short video tutorial together to guide first time users through Lovely Charts main features in 90 seconds. Click on the button below to have a look at it, or dive in directly if you prefer by selecting a chart type in the list on the right.

**Create new**

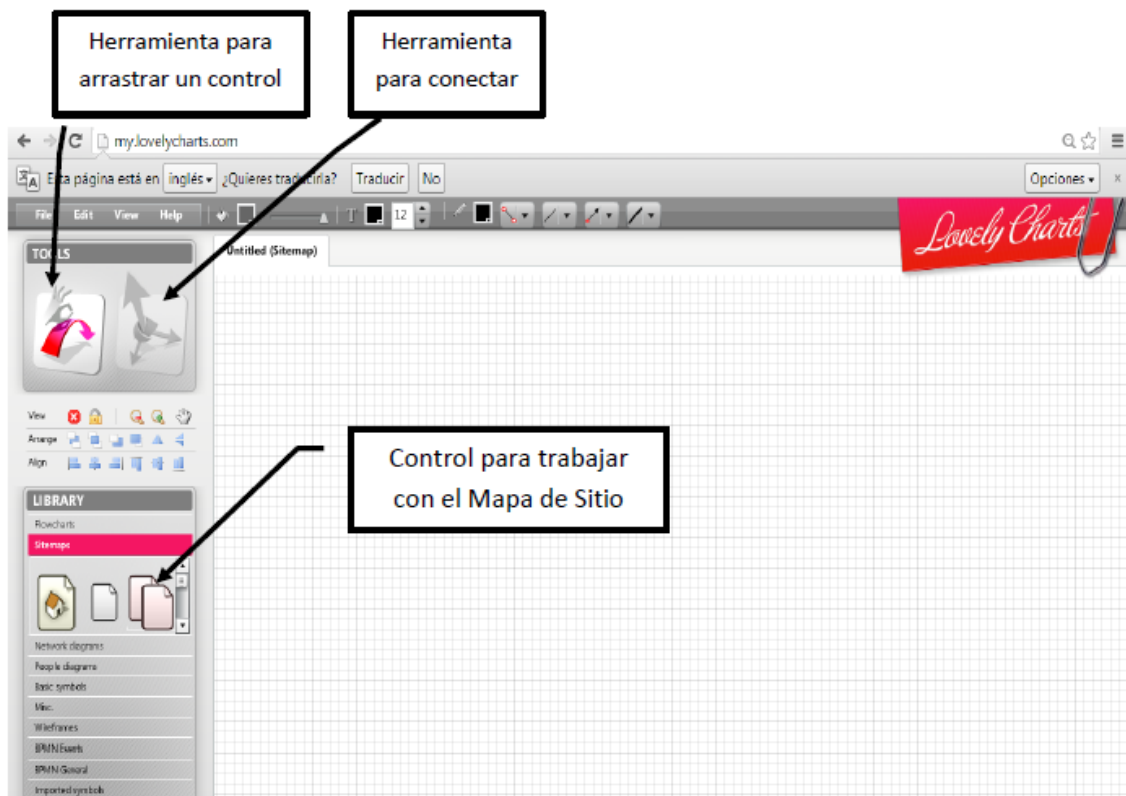
- Flowcharts
- SiteMaps
- Network diagrams
- People diagrams
- Basic symbols
- Misc.
- Wireframes
- BPMN Events
- BPMN General
- Imports

**Upgrade your account NOW**

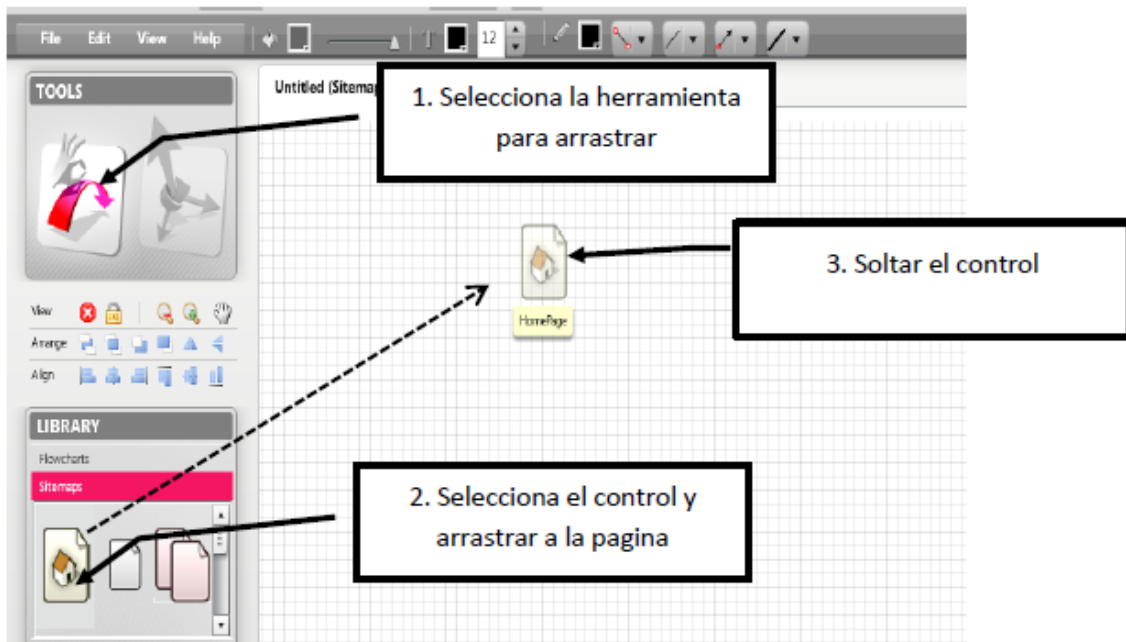
**Annotation:**

- Selecciona la opción para crear un Mapa de Sitio (SiteMaps) (points to SiteMaps in the list)

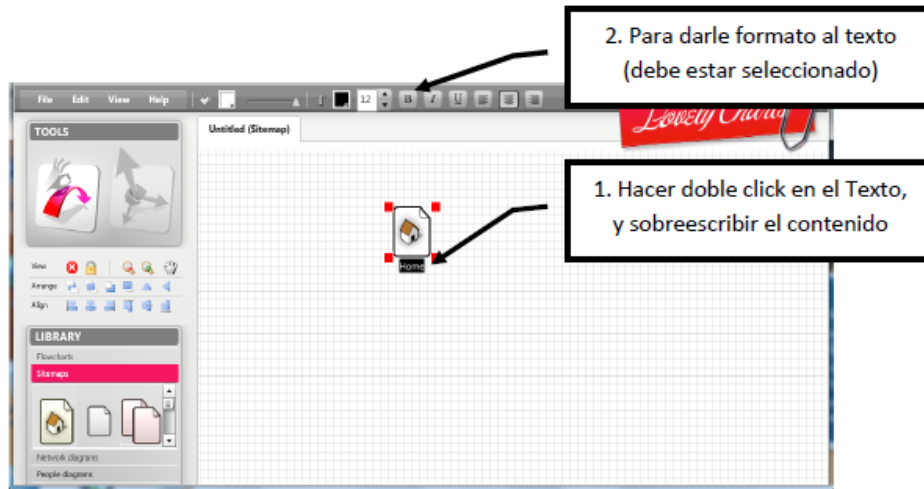
A continuación la plataforma carga el aplicativo para crear mapa de sitios, tal como se muestra



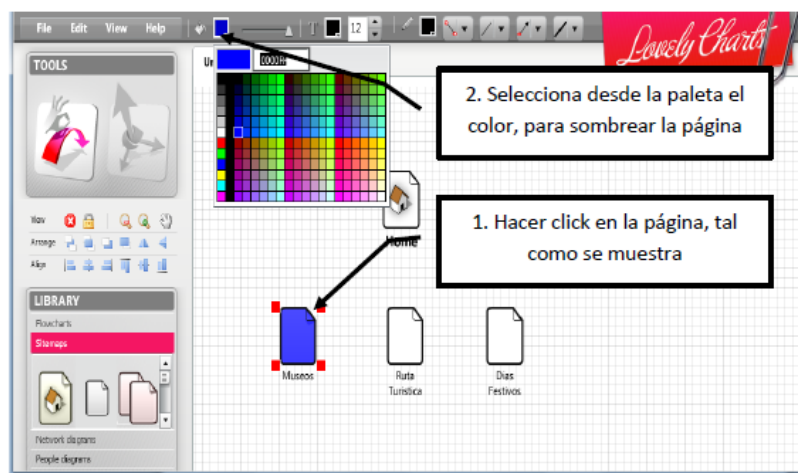
Para insertar una objeto a la plantilla, siga los pasos que se muestra a continuación.



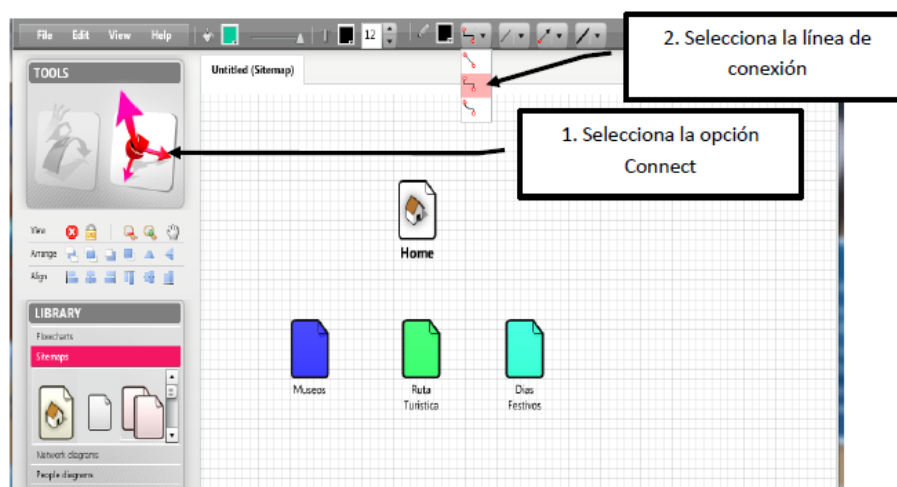
A continuación escribimos un texto al objeto incrustado en la plantilla.



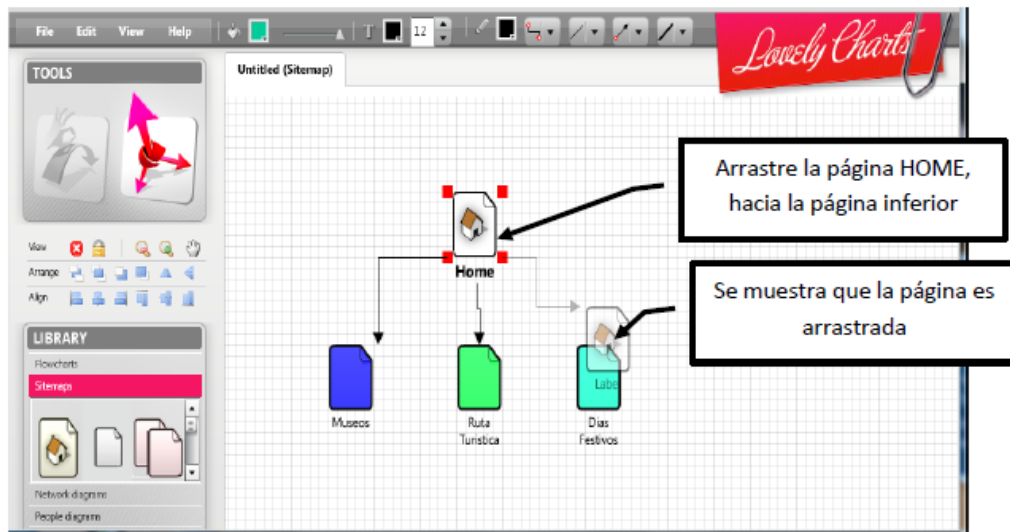
Asignar colores de fondo a las páginas, para resaltar su presentación en el mapa de sitio



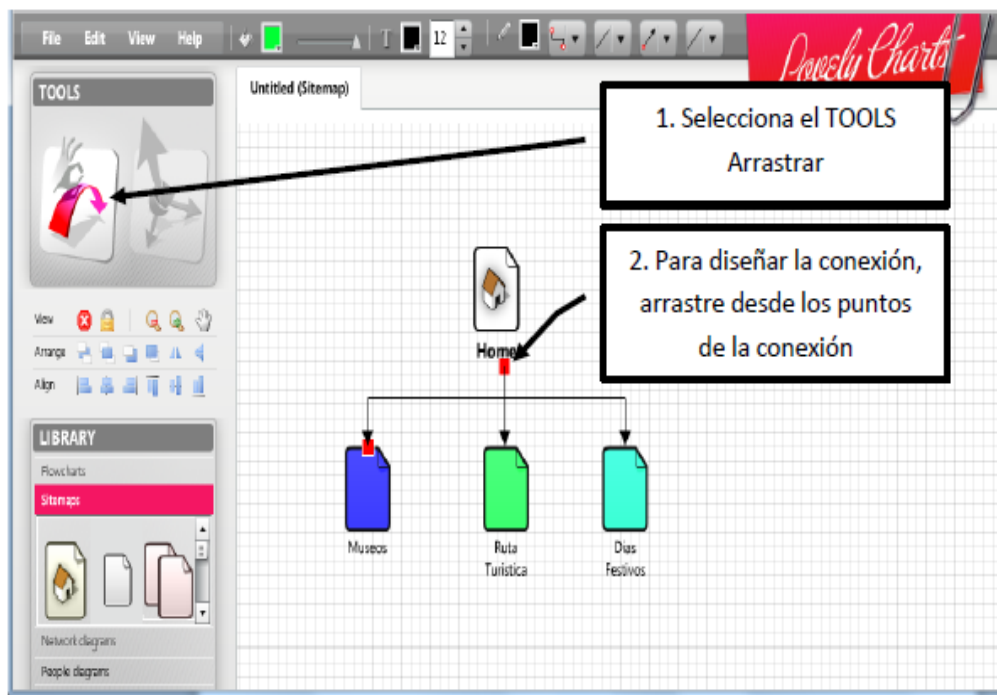
Conexión entre páginas: 1. Selecciona desde la opción TOOLS, la opción conect.  
2. Selecciona la línea de conexión, tal como se muestra



Para realizar la conexión entre las páginas, arrastre la página HOME, hacia las páginas que se encuentra en la parte inferior, tal como se encuentra, las cuales se encontrarán conectadas.

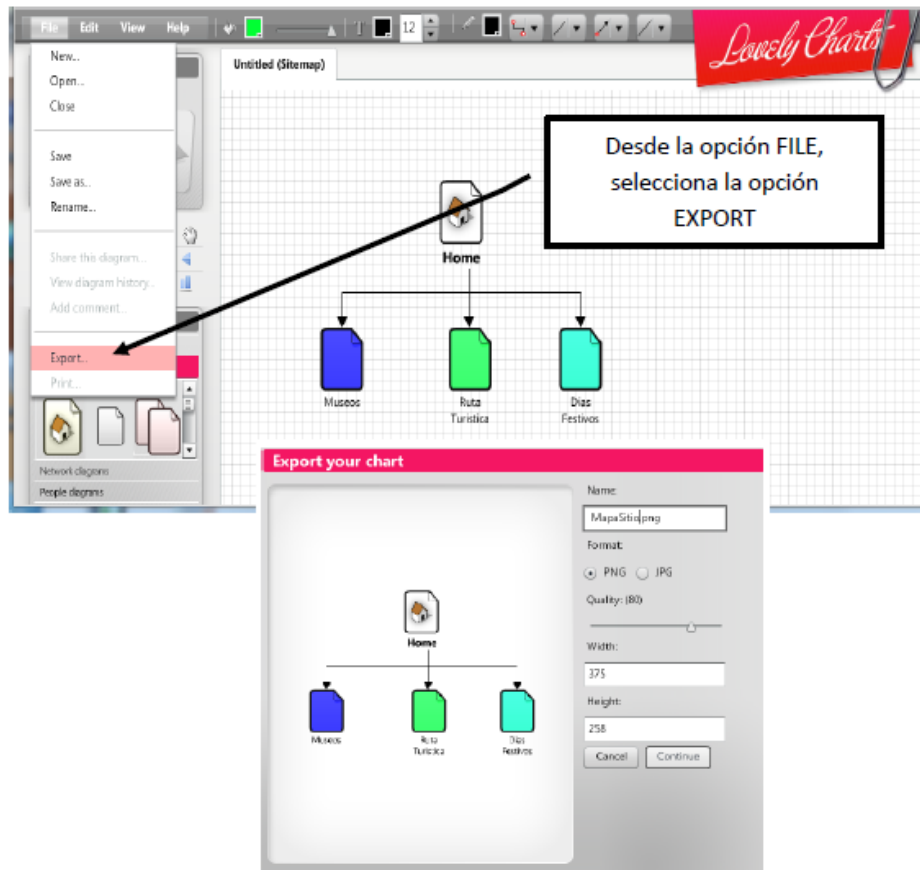


Para diseñar la conexión lineal: 1. Selecciona el TOOLS la opción Arrastrar; 2. Las líneas de conexión se arrastrarán por los extremos (puntos rojos)

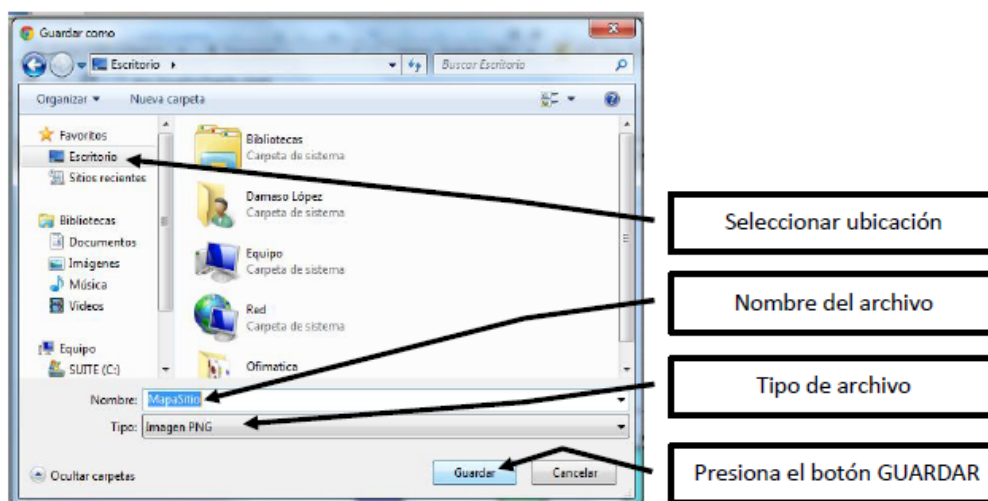


## Exportando el mapa de Sitio

Seleccionar la opción Export, donde se visualiza una ventana. Indicar el nombre del archivo y seleccionar su formato: PNG o JPG y presionar el botón **CONTINUE**



A continuación se muestra la ventana de diálogo para guardar el archivo. Ingresar el nombre de la imagen y presionar el botón GUARDAR





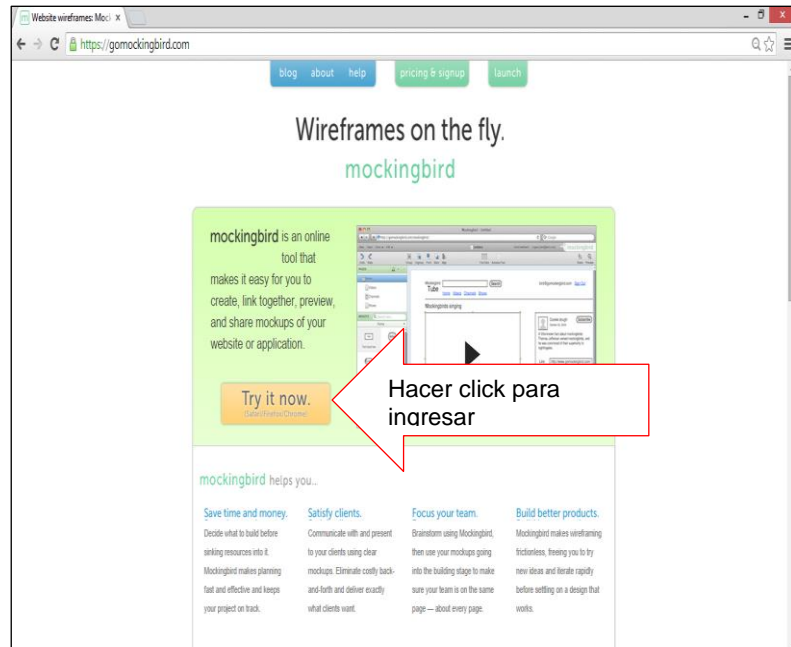
## LABORATORIO 1.3

### Diseñando un WireFrame con MOCKINGBIRD

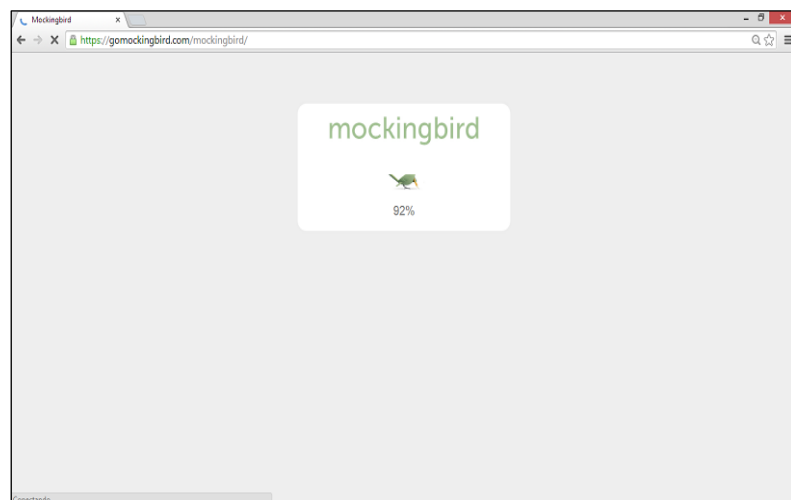
Se pide diseñar un WIREFRAME utilizando el aplicativo MOCKING BIRD

Para ingresar al Mockingbird:

Desde la URL  
ingresamos la  
siguiente dirección:  
<https://gomockingbird.com/>

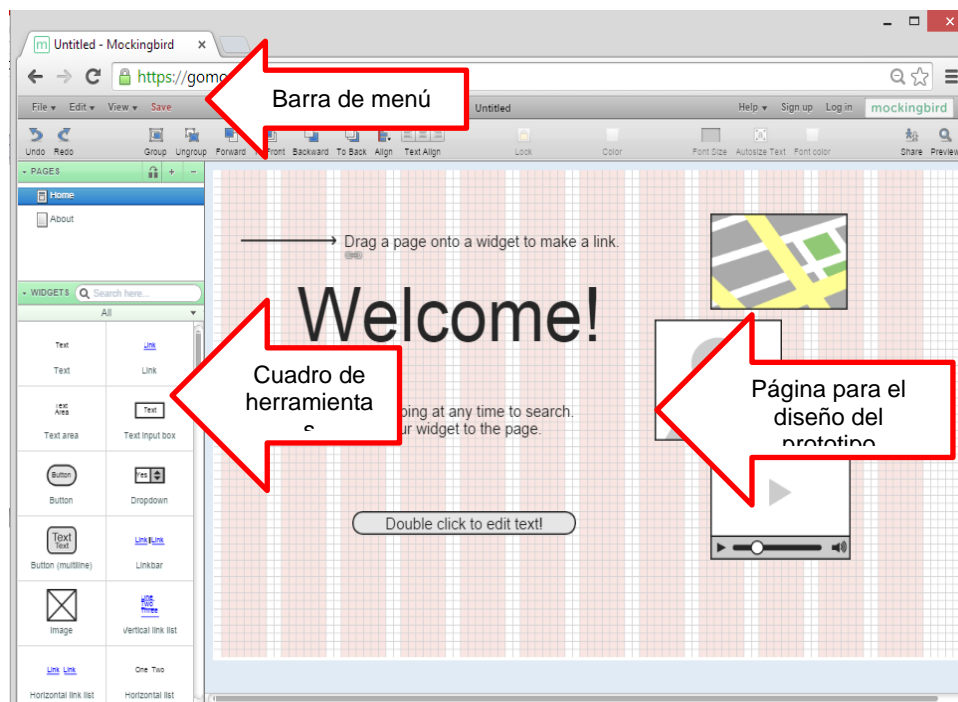


Cargando la página  
MockingBird.



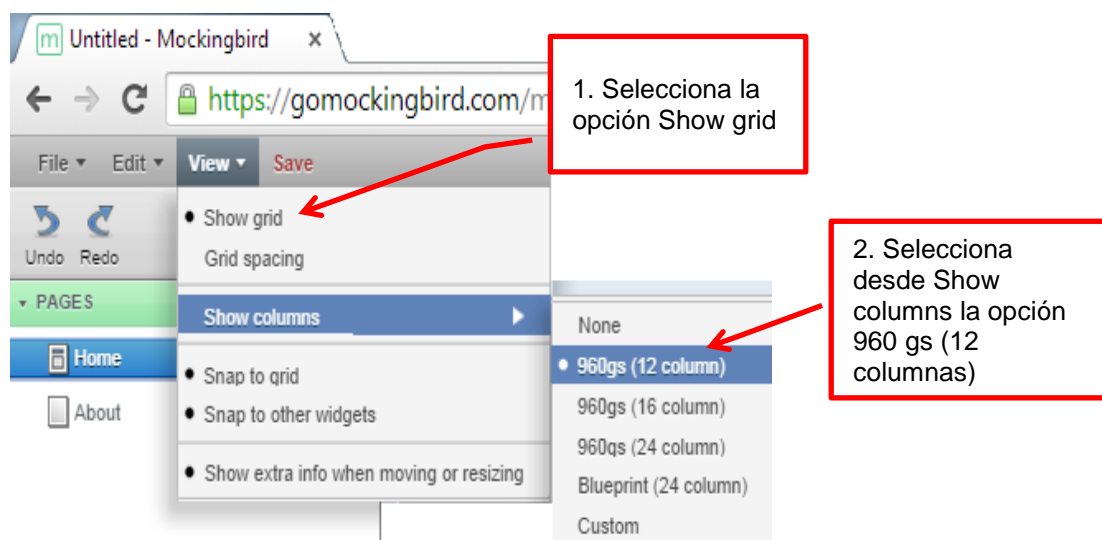
## IDE del MockingBird

Al cargar el aplicativo se visualiza el siguiente IDE, tal como se muestra

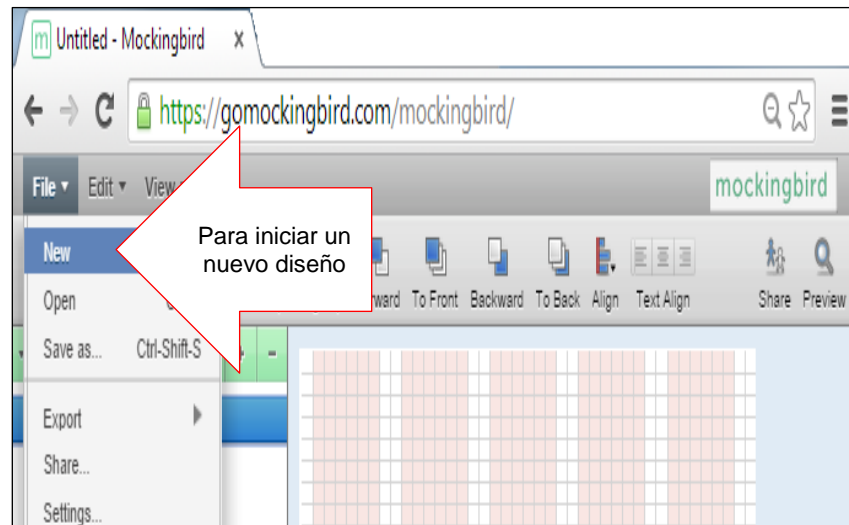


Para configurar la vista del wireframe en cuadrículas, desde la opción VIEW:

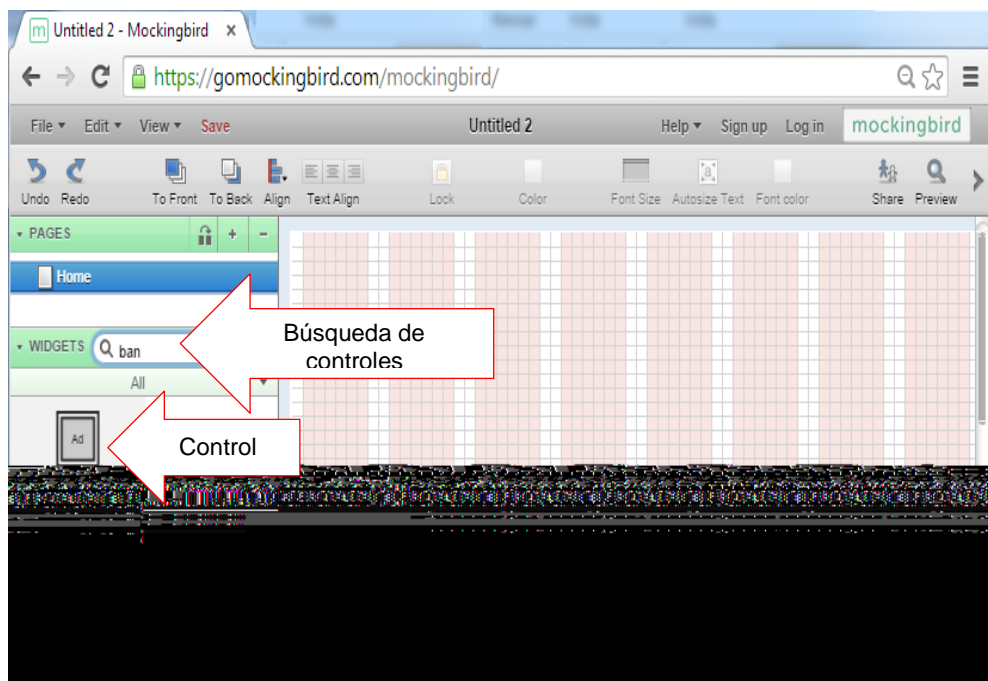
1. selecciona la opción Show grid
2. selecciona desde Show columns la opción 960gs (12 columns), para visualizar 12 columnas en la cuadrícula



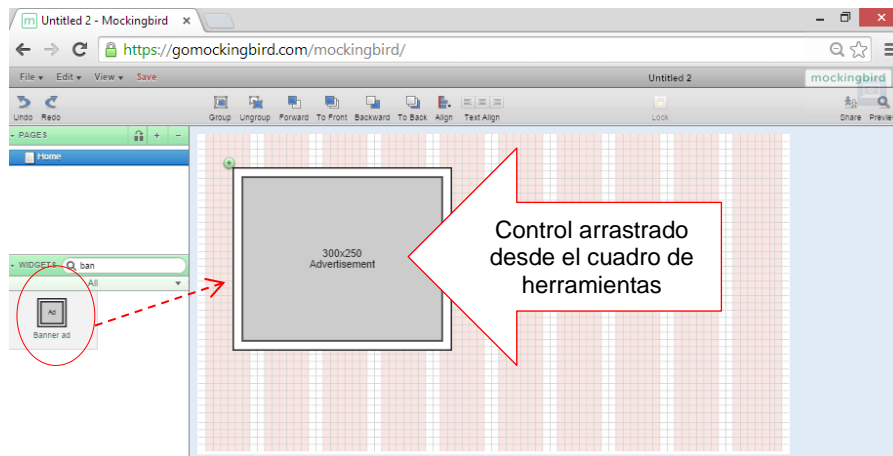
Para iniciar un nuevo diseño, selecciona, desde la opción FILE, la opción New, tal como se muestra.



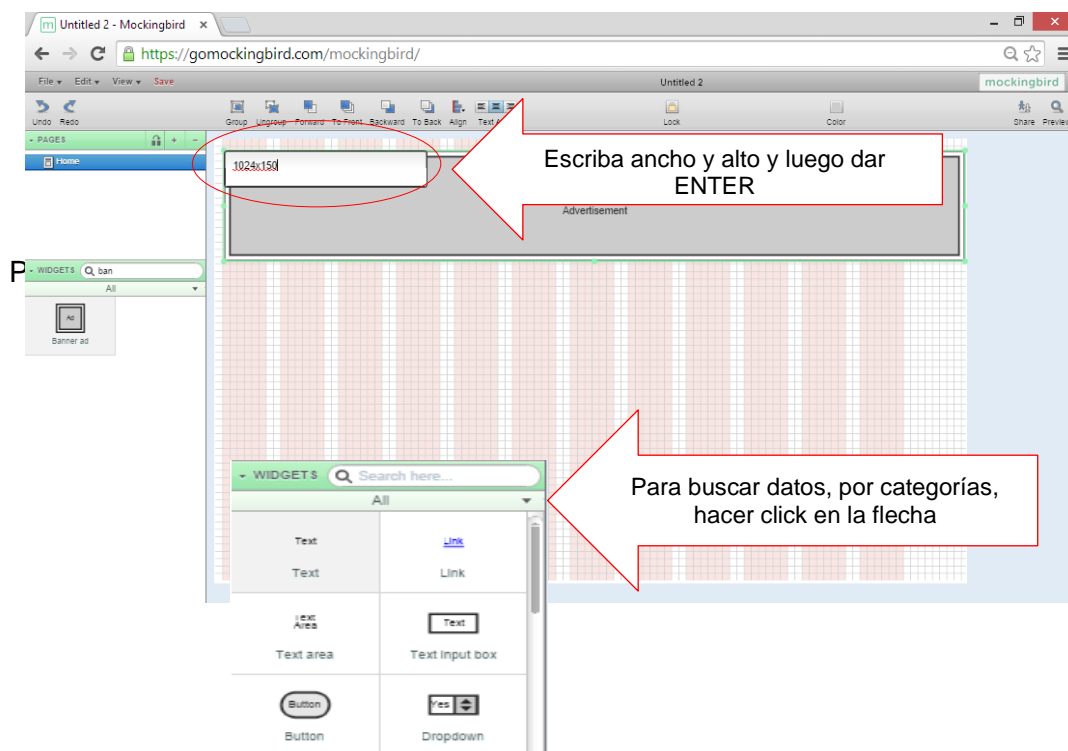
Para la búsqueda de un control, escriba en la opción WIDGET, las iniciales del nombre del control, y el buscador mostrará los controles que coincidan con la búsqueda, tal como se muestra en la figura.



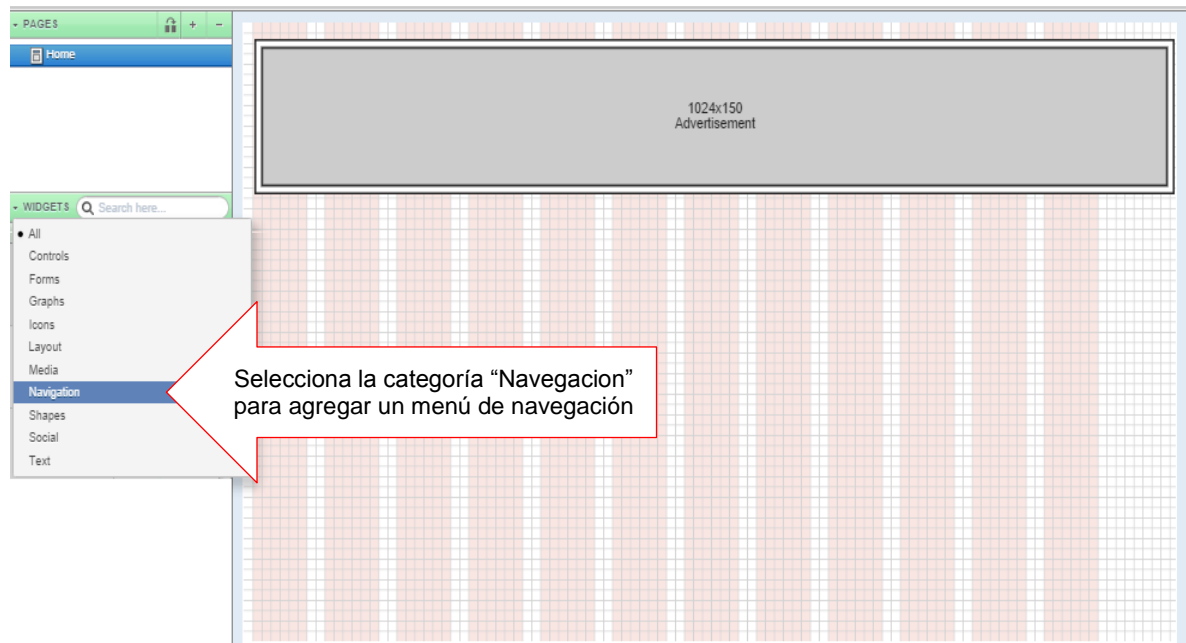
Para colocar un control a la grilla del wireframe, debemos arrastrarlo a la página.



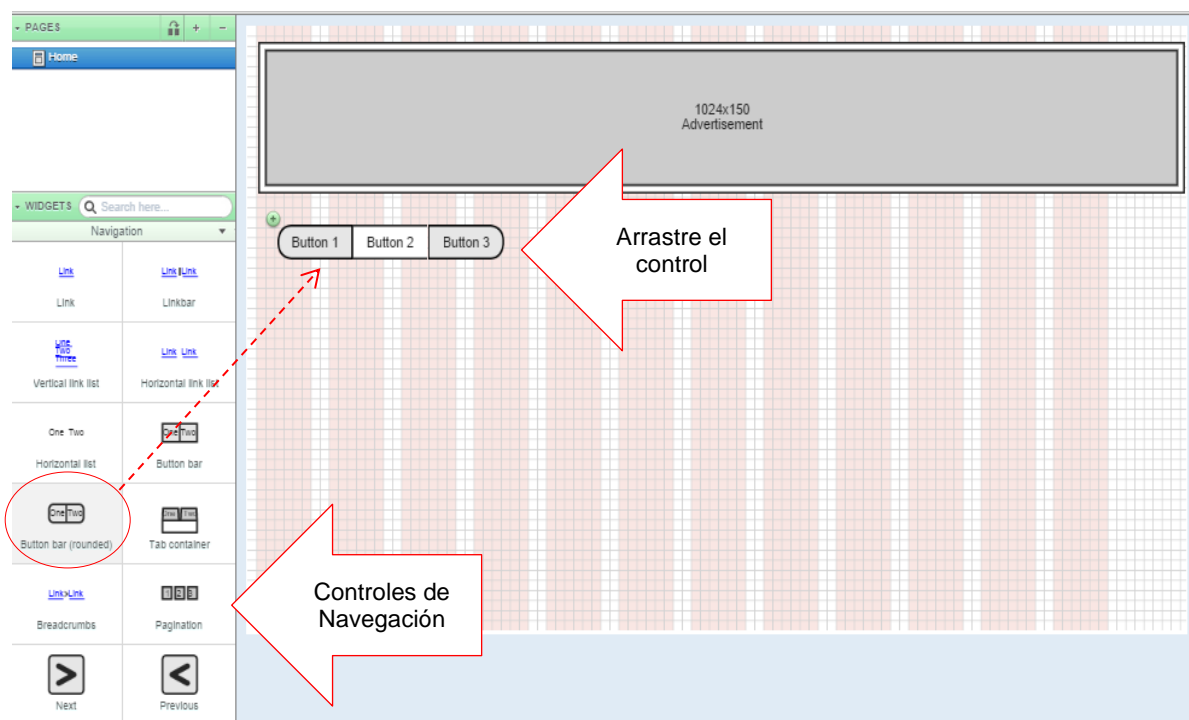
Ubicar y dimensionar el control agregado. Para dimensionar el control, hacer doble click sobre dicho control y definir su ancho y su alto (**anchoxalto**), tal como se muestra.



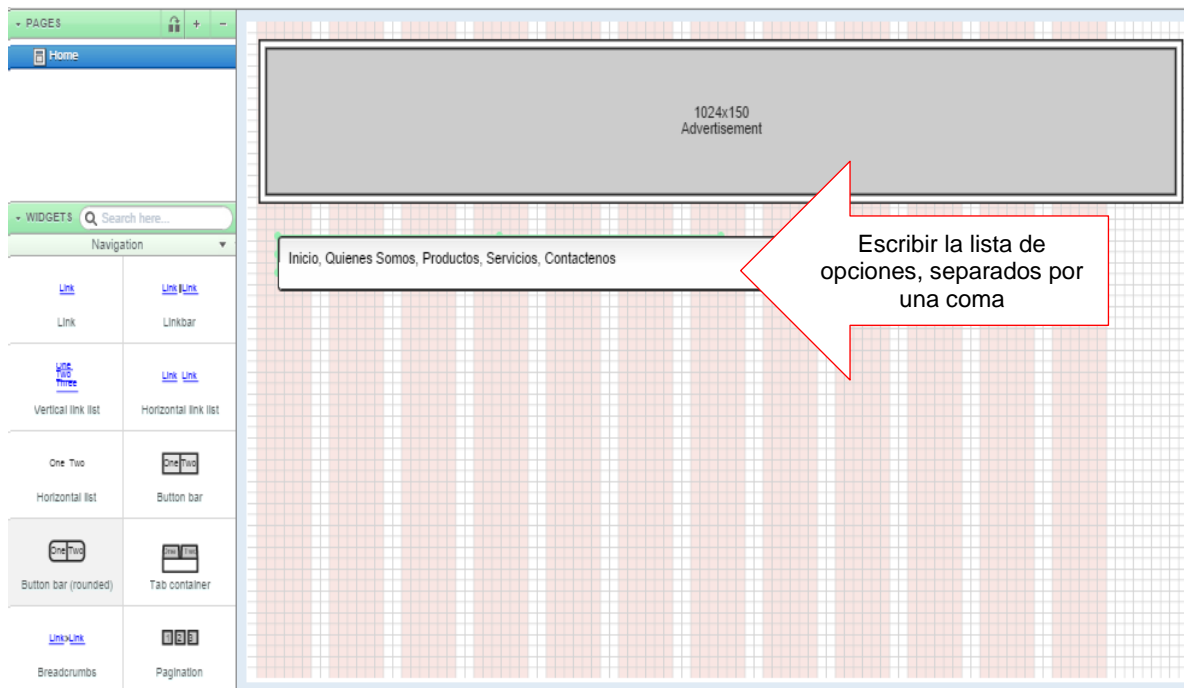
Si queremos agregar un menú de navegación, selecciona la categoría NAVEGACION, tal como se muestra.



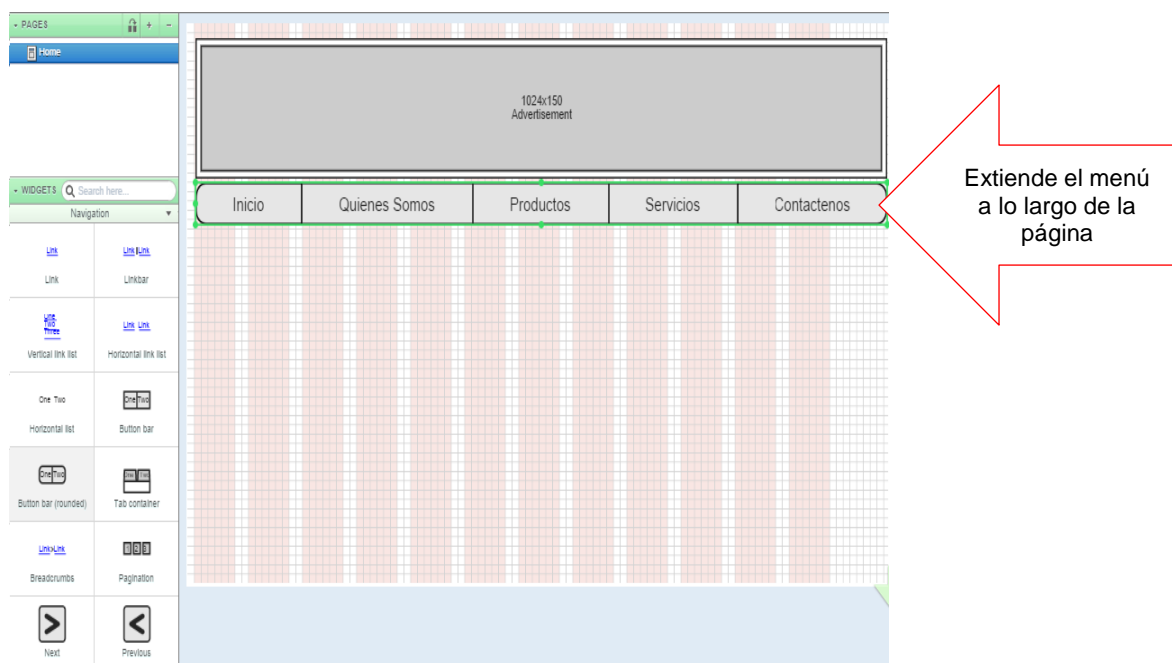
Arrastre, desde el cuadro de herramientas, el control (Button bar) Barra de menú, tal como se muestra.



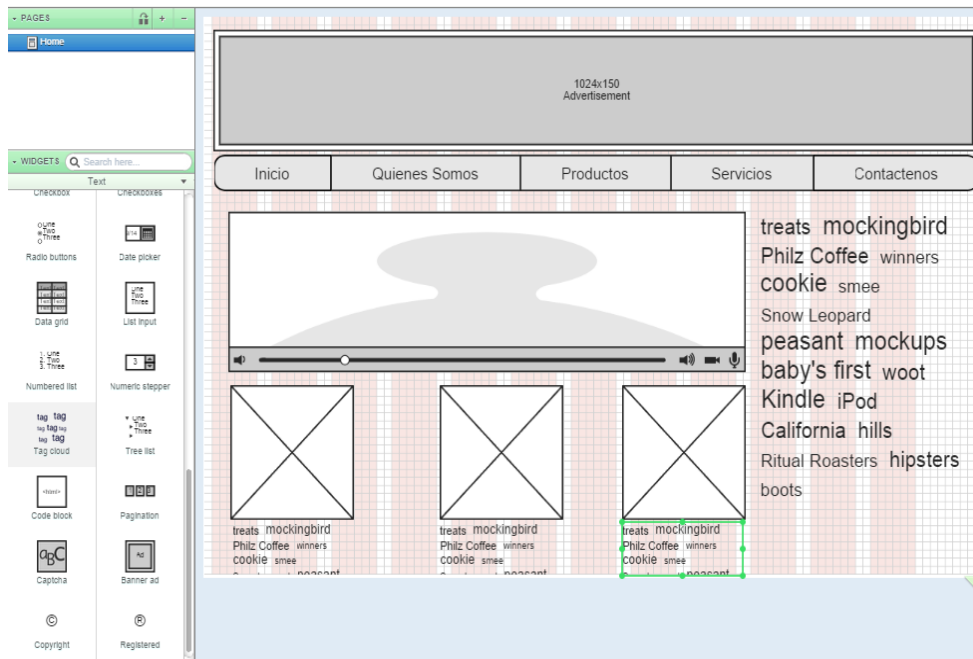
Una vez que lo sueltas, se edita el menú bar. A continuación escribimos la lista de los elementos, los cuales se separan por una coma, tal como se muestra. Al finalizar presionar ENTER.



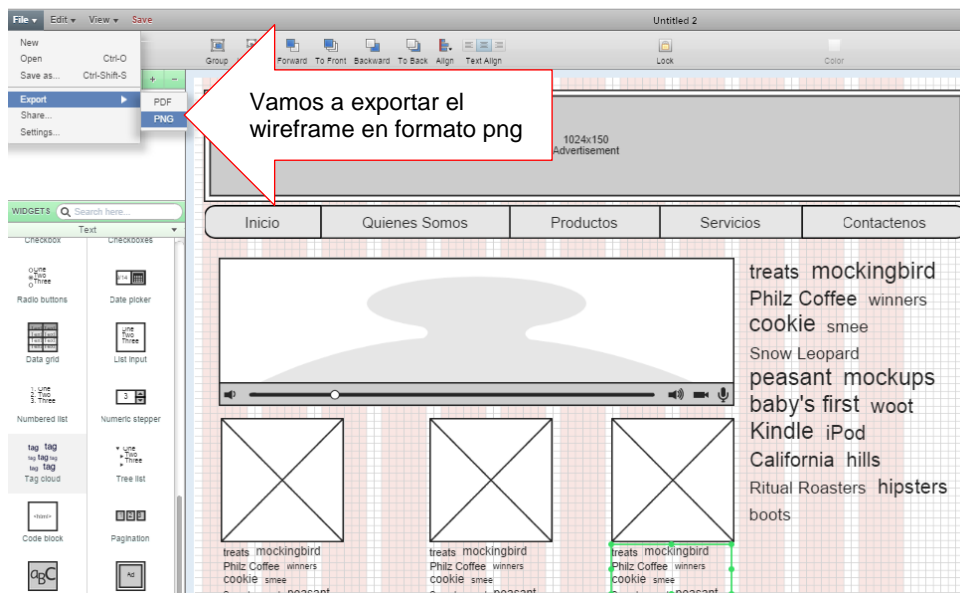
Extienda el control a lo largo de la página, diseñado el menú se visualiza de esta forma



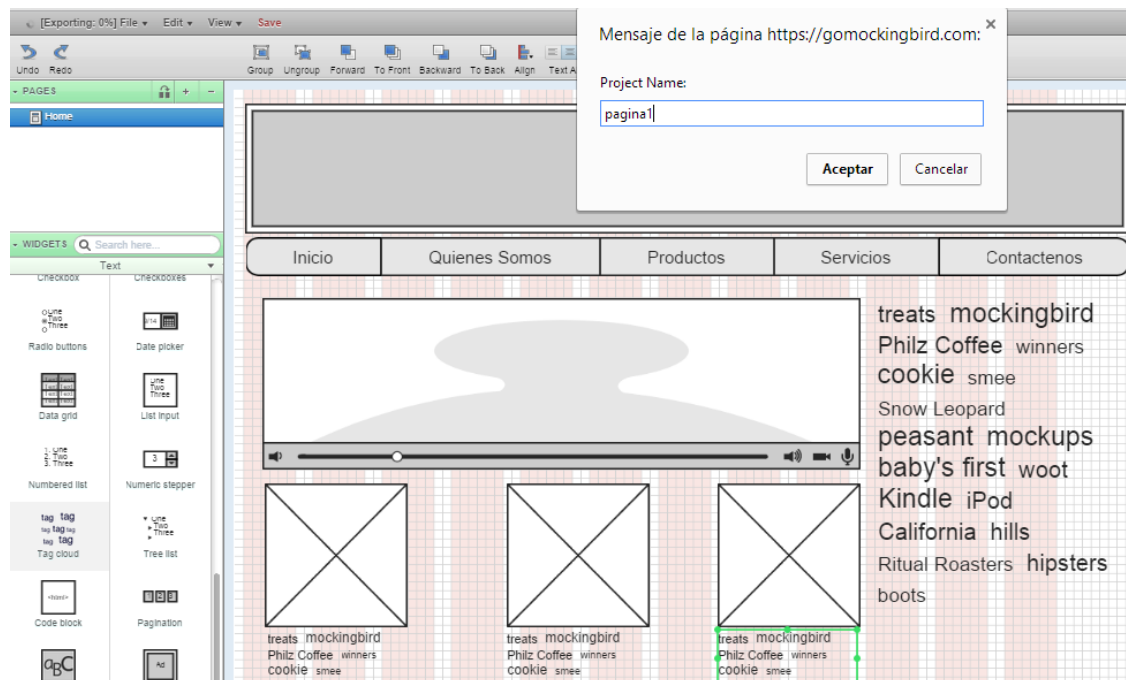
A continuación termine el diseño del wireframe tal como se muestra



Al finalizar exportar el WireFrame, desde la opción FILE, selecciona la opción EXPORT en formato PNG



A continuación el proyecto solicita el nombre del proyecto para descargarlo a tu pc.





# Resumen

- 📖 Internet se le define como una red mundial de computadoras, distribuidos por el todo el mundo, las cuales se encuentran interconectadas a través de satélites, líneas telefónicas, fibra óptica, etc., donde utilizando un protocolo de comunicaciones común (TCP/IP), que permite a millones de usuarios estar conectados para compartir, intercambiar, publicar y extraer información.
- 📖 Sitio Web, Se le conoce como **Site**, un sitio web es un conjunto de páginas web, relacionadas a un dominio de Internet el cual se puede visualizar a través de internet.
- 📖 Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.
- 📖 El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.
- 📖 Un dominio de Internet es un nombre de un sitio web. Cada dominio tiene que ser único en Internet, por ejemplo “elcomercio.pe” es el nombre del dominio de la página web del diario El Comercio
- 📖 Un navegador o navegador web (del inglés, web browser) es un programa que permite visualizar la información que contiene una página web la cual puede estar alojada en un servidor dentro de la World Wide Web o en uno local.
- 📖 Etapa de un proyecto web permite desarrollar un sitio web efectivo requiere una planeación bien elaborada. Puede ahorrar tiempo y dinero durante el proceso del proyecto si logra entenderlo en su totalidad antes de comenzar. A pesar de que los sitios varían en rango de contenido, desde unas cuantas páginas hasta sitios mucho más complejos con características sofisticadas, el proceso de desarrollo se mantiene igual.
- 📖 Arquitectura web es una tarea que requiere conocimientos técnicos de construcción, funcionales y de diseño para sitios o páginas web. Con base en las preferencias del contenido se crea un mapa de la arquitectura del sitio para demostrar visualmente cómo se organizan el contenido y la estructura del sitio.
- 📖 Un Sitemap o Mapa del Sitio es la estructura de enlaces de páginas web que definirá la buena o mala navegación de un sitio web.
- 📖 Un wireframe básicamente es un boceto básico y de baja calidad del desarrollo de una página web o el diseño de una interfaz, cuya finalidad es el mostrar al cliente un boceto rápido para facilitar la comunicación entre cliente y desarrollador.
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 [http://www.hipertexto.info/documentos/internet\\_tegn.htm#IP](http://www.hipertexto.info/documentos/internet_tegn.htm#IP)

🔗 [http://docentesistemas.webnode.es/contenidos-programaticos/grado-11/pagina-web-generalidades/que-es-un-hosting-/](http://docentesistemas.webnode.es/contenidos-programaticos/grado-11/pagina-web-generalidades/que-es-un-hosting/)

🔗 <http://es.slideshare.net/josegregoriob/servidor-web-8451426>

🔗 <http://www.servicio-hosting.com/tipos-de-web-hosting.php>

🔗 <http://www.duplika.com/que-es-el-hosting>

- ✎ <http://culturacion.com/como-escoger-un-servidor-para-alojar-mi-sitio-web/>
- ✎ <http://www.quees.info/que-es-un-dominio.html>
- ✎ <http://www.tiposde.org/internet/113-tipos-de-navegadores/>
- ✎ <http://spn314.blogspot.com/2013/11/cuantos-navegadores-de-internet-hay.html>
- ✎ <http://www.condo-consulting.com/test-de-usabilidad-que-son-y-para-que-me-sirven.b.aspx>
- ✎ <http://disenowebakus.net/pasos-clave-en-la-creacion-de-un-sitio-web.php>
- ✎ <http://novatosweb.blogspot.com/2010/12/etapas-que-debe-formar-tu-proyecto-web.html>
- ✎ <http://es.slideshare.net/cilp2884/etapas-de-un-proyecto-web>
- ✎ <http://es.slideshare.net/christian.bonner/arquitectura-de-la-informacin-de-sitios-web>
- ✎ <http://tednologia.com/que-es-un-mockup-o-wireframe/>
- ✎ <http://olgacarreras.blogspot.com.es/2007/02/wireframes.html>



# FUNDAMENTOS DE UN PROYECTO WEB Y ELEMENTOS GRAFICOS

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno entiende el concepto de un proyecto web y diseña los elementos gráficos necesarios para preparar un proyecto web.

## TEMARIO

### 1.1. Tema 1: Fundamentos de un proyecto web

- 1.1.1 Generalidades
  - 1.1.1.1 Internet
  - 1.1.1.2 Servidores web
  - 1.1.1.3 Hosting y dominios
  - 1.1.1.4 Navegadores
- 1.1.2 Etapas de un proyecto web
  - 1.1.2.1 Arquitectura de un proyecto web
  - 1.1.2.2 Mapa de sitio
  - 1.1.2.3 Wireframes.

### 1.2. Tema 2: Elementos gráficos para la web

- 1.2.1 Manejo de un editor de imágenes
  - 1.2.1.1 Tipos de imagen, resolución, modos de color.
  - 1.2.1.2 Tamaño, recorte, selección, etc.
- 1.2.2 Creación de banners, menús y botones.

## ACTIVIDADES PROPUESTAS

- Los alumnos desarrollan los laboratorios de esta semana
- Los alumnos desarrollan diseñan el mapa de sitio de su proyecto
- Los alumnos desarrollan diseñan su wireframe de baja y alta fidelidad.



## 1.2. ELEMENTOS GRAFICOS PARA LA WEB

Los elementos gráficos son una parte fundamental del diseño en todas sus modalidades, introduciendo en las composiciones información visual que complementa en gran medida el mensaje que se desea transmitir, y que a veces es tan importante en el diseño como los contenidos textuales del mismo.

A la hora de realizar la ilustración de páginas web, podemos optar bien por dibujarlas a mano, escanearlas y retocarlas si es preciso, bien por empezar el dibujo directamente en un programa gráfico adecuado.

Los programas gráficos son de gran ayuda a la hora de trabajar con ilustraciones, ya que en ellos se realizan fácilmente operaciones que en dibujo o pintura son muy laboriosas (tareas repetitivas, copiar una imagen entera o parte de ella, dibujar trazados geométricos perfectos, aplicar filtros o efectos, etc.).



Figura 1

Referencia: <https://play.google.com/store/apps/details?id=com.creapp.photoeditor>

Para crear una ilustración podemos, en principio, usar cualquier programa gráfico, aunque dependiendo del tipo de ilustración buscada habrá unos más adecuados que otros. Así, si en nuestra ilustración van a predominar los trazos limpios y geométricos resultan especialmente indicados los programas de gráficos vectoriales, como Adobe Illustrator o Macromedia Freehand. En cambio, si lo que buscamos es una ilustración que parezca más natural, con trazos difuminados, juegos de colores o luces, degradados o tonos concretos, serán más indicados programas de retoque, como Adobe Photoshop o Paint Shop Pro.

### 1.2.1. MANEJO DE UN EDITOR DE IMAGENES

Los editores de imágenes son aplicaciones informáticas que permiten el tratamiento y manipulación de imágenes, fotografías, vídeos, etc., de la forma más simple (reducir, ampliar, retocar, recortar...) hasta la más

compleja (hacer montajes y arreglos, animar vídeos y figuras...) para mejorar su presentación.



**Figura 2**

Referencia: <http://mashable.com/2010/07/15/wireframing-tools/>

Una de las metas puede ser eliminar las fallas que pueden haberse producido durante el escanéo o al fotografiar, por ejemplo sobre exposición, bajaexposición, falta de contraste, ruido en la imagen, efecto de los ojos rojos, paradoja de las líneas paralelas en perspectiva, etc. Estas fallas se producen por defectos técnicos en los aparatos fotográficos, escáner, condiciones de trabajo precarias, fallas en la operación u originales defectuosos.

Un editor de gráficos vectoriales es una aplicación informática que permite al usuario crear y editar imágenes de graficos vectoriales (imagen digital formada por objetos geométricos independientes, cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc.), de forma interactiva en la pantalla de la computadora y guardarlas en uno de los tantos formatos de graficos como pdf. Entre los editores vectoriales tenemos:

- Adobe Illustrator
- Corel Draw
- Inkscape
- OpenOffice.orgDraw

Y de editors de mapa de bits es una aplicación informática que permite al usuario crear y editar imágenes de mapa de bits (imagen que esta formada por pixeles). Entre los editores de mapa de bits tenemos;

- Adobe Photoshop
- Paint
- Gimp
- FireWork

### 1.2.1.1 Tipos de imagen, resolución, modos de color

#### 1. Tipos de imagen

A la hora de trabajar con imágenes se pueden distinguir dos tipos de imágenes. Por una parte, se encuentran las imágenes vectoriales, las cuales se componen de líneas y curvas definidas matemáticamente, y por otra las imágenes de mapa de bits que están compuestas por una rejilla o mapa de pequeños cuadrados denominados píxeles.

Los programas de dibujo generan imágenes de tipo vectorial. A veces se llaman "programas de dibujo orientado a objeto", ya que las imágenes comprenden objetos independientes, líneas definidas matemáticamente y figuras.

Los programas de pintura generan imágenes de tipo mapa de bits, si se dibuja una línea, la aplicación la convierte en unos pequeños puntos cuadrados llamados píxeles.

Entre los formatos de archivos de imagen tenemos:

- BMP. Formato nativo de Paint de Microsoft y soportado por gran cantidad de aplicaciones Windows y DOS.
- GIF. de CompuServe, utilizado en los viejos tiempos para la compresión de archivos.
- PCX. Formato nativo de PC PaintBrush.
- PDF. Variación del lenguaje de impresión PostScript.
- JPEG. Formato de compresión más eficiente y esencial actualmente disponible.
- TIFF. Formato de impresión de imágenes ampliamente soportado tanto en Macintosh como en PC.

## **2. Resolución**

El término resolución se refiere al número de unidades de imagen por pulgada. En los monitores y los escáneres estas unidades se llaman píxeles; en impresoras y fotocomponedoras se denominan puntos.

Cuando se decide la resolución necesaria para escanear una imagen, el objetivo es reunir suficiente información para que se vea bien, pero de tal modo que el archivo no sea tan grande que no se pueda trabajar.

## **3. Modos de color**

La propiedad más básica de una imagen es su modo. Hay tres modos posibles: RGB, escala de grises e indexado. RGB son las iniciales en inglés de rojo, verde y azul, e indica que cada punto de la imagen está representado por un nivel de "rojo", un nivel de "verde" y un nivel de "azul", representando una imagen a todo color. Cada canal de color tiene 256 niveles de intensidad posibles. Para más detalles, consulte modelos de color. En una imagen de escala de grises, cada punto está representado por un valor de brillo, que va desde el 0 (negro) hasta el 255 (blanco), con valores intermedios que representan diferentes niveles de gris.

### **1.2.1.2 Tamaño, recorte, selección, etc.**

#### **1. Tamaño**

Cuando modificas el tamaño de una foto, creas una copia nueva con una resolución de imagen ajustada (número de píxeles) y con todos los cambios aplicados.

La foto original del disco duro permanece intacta en todo momento.

Puedes modificar el tamaño de las fotos al exportarlas, enviarlas por correo electrónico o almacenarlas.

## 2. Recorte

Hay muchas razones para recortar una imagen; por ejemplo, ajustar una imagen para rellenar un marco, quitar una parte del fondo para resaltar el sujeto, etc. Para realizar este proceso, debes seleccionar la herramienta recorte y abrir la foto para iniciar el proceso. Al terminar el proceso, la sección recortada podrá ser almacenada en otra página.

## 3. Selección

Todo editor de imágenes tiene una herramienta para seleccionar partes concretas de la imagen, sobre la cual actuaremos posteriormente con otras herramientas. Las herramientas de selección pueden ser de marco rectangular, elíptico, fila y columna.

## 4. Varita Mágica

Es una opción donde podremos realizar selecciones basándonos en la similitud de color de los píxeles próximos al lugar pulsado. Es decir, donde se pulse con la herramienta, quedará seleccionada toda la zona próxima a ese color.

### 1.2.2 Creación de banners, menús y botones.

Crear un banner o logo con un editor de imágenes es una tarea sencilla, incluso se pueden utilizar herramientas muy básicas como Paint y luego mejorar su diseño con otros programas como por ejemplo el Photoshop.



Figura 3

Referencia: <http://www.todopatuweb.net/news/como-crear-un-banner-de-publicidad-para-tu-web/>

Lo cierto es que si quieres que tus banners y logos cuenten con efecto flash, deberás utilizar otras herramientas algo más complejas o recurrir a los generadores online.

Si optas por crear tus logos y banners con alguno de los programas antes mencionados, solo deberás tener en cuenta las medidas, las cuales suelen ser estándar. Los generadores de banners y logos online precisamente se basan en dichas dimensiones, verás que se muestran entre las opciones de fuente y selección de colores.



Un editor de imágenes dispone de un Editor de botones el cual puede crear botones personales arrastrando formas, importando imágenes de gráficos o arrastrando objetos en la ventana de documento. Después el Editor de botones le guiará por los pasos que permiten controlar el comportamiento del botón.



Figura 4

Referencia: [aprendeenlinea.udea.edu.co/lms/ova/mod/resource/view.php?id=1559](http://aprendeenlinea.udea.edu.co/lms/ova/mod/resource/view.php?id=1559)

Una barra de navegación, también denominada bar nav, es un grupo de botones que ofrece vínculos a otras áreas de un sitio Web.

Normalmente, no cambia de aspecto en el sitio, de modo que ofrece un método coherente de navegación sin importar el lugar en que se encuentre el usuario en el sitio Web. La barra de navegación tiene el mismo aspecto de una página Web a otra, pero los enlaces pueden ser específicos para el funcionamiento de cada página.

En un editor de imágenes como Fireworks, puede crear un bar nav creando un símbolo en el Editor de botones y luego poniendo instancias de dicho símbolo en el lienzo.

## LABORATORIO 2.1

### Trabajando con FIREWORKS

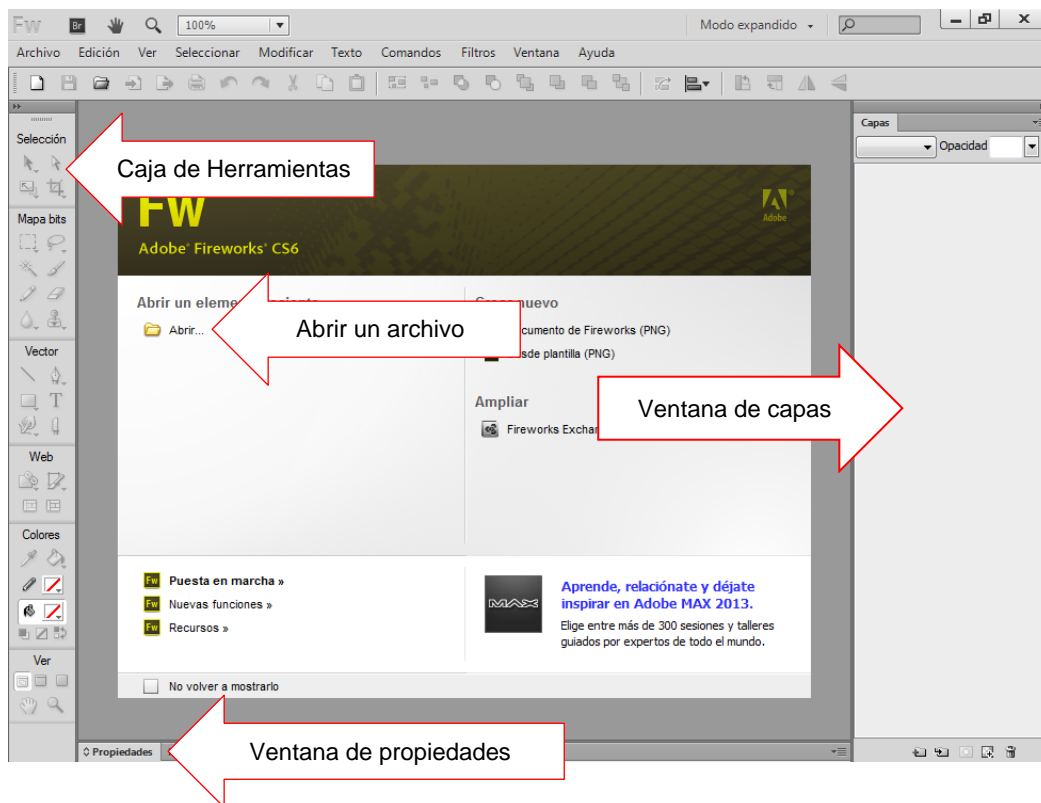
Se pide diseñar un wireframe de alta fidelidad utilizando el aplicativo FireWorks.

#### Pasos a seguir:

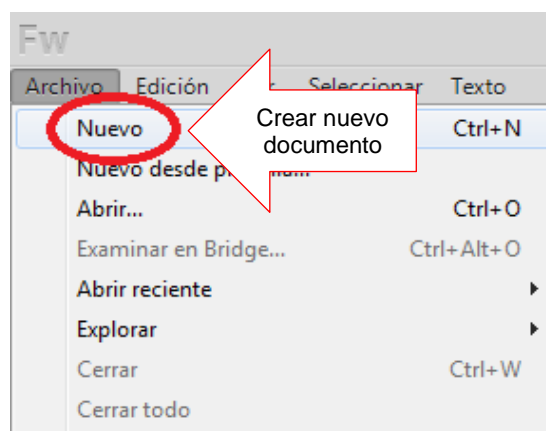
Para crear un sitio Web, inicie el aplicativo FIREWORKS



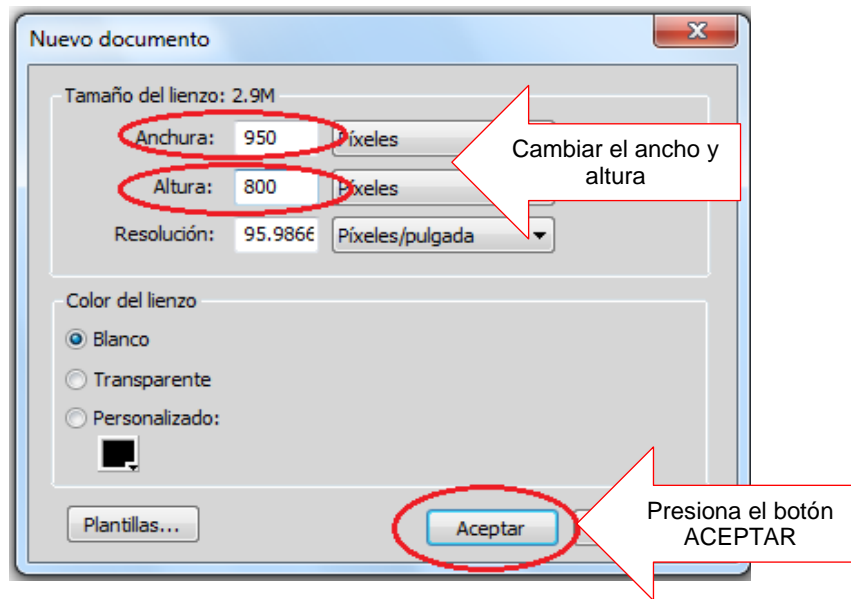
A continuación se muestra el siguiente IDE.



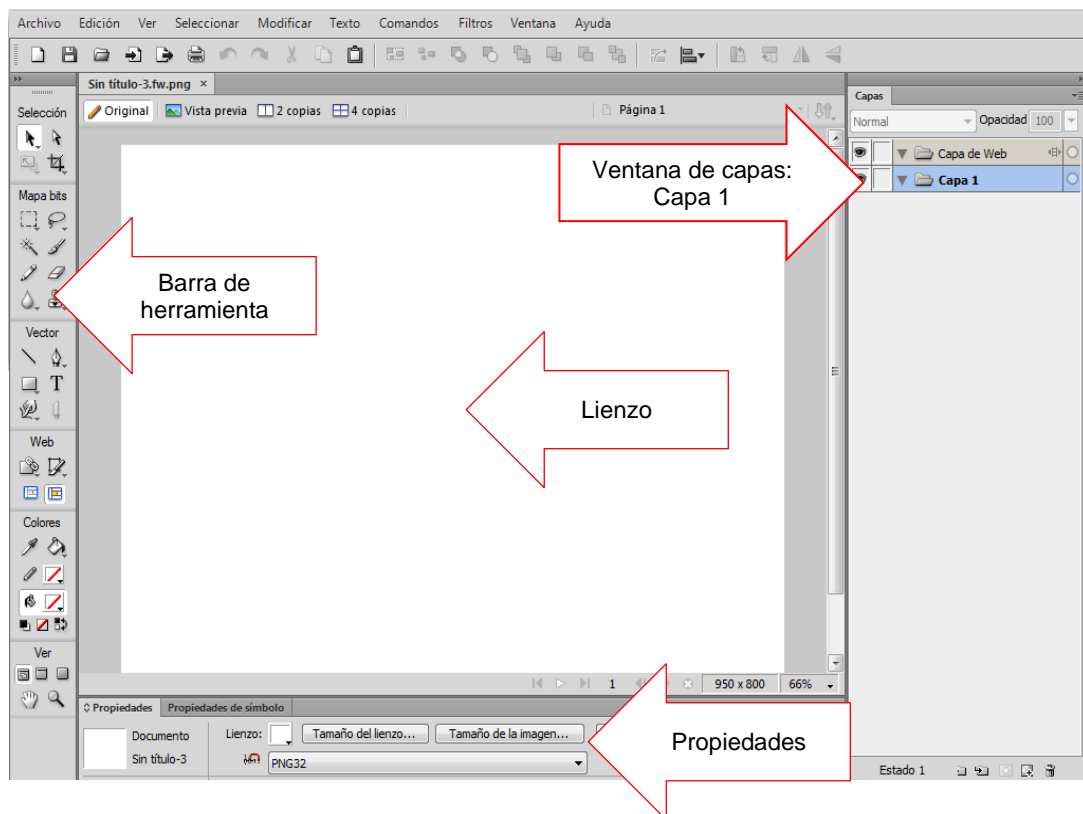
Para iniciar con el diseño del wireFrame de alta fidelidad, primero creamos un nuevo documento. Hacer click en la opción Nuevo, para crear un documento



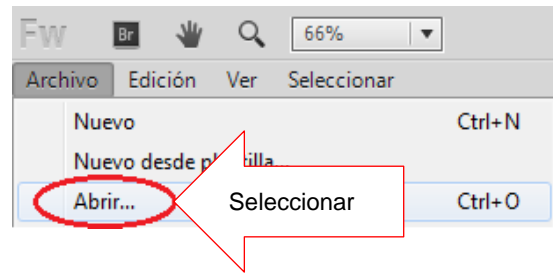
En el nuevo documento, configurar el tamaño del lienzo, tal como se muestra. Al finalizar presiona el botón ACEPTAR.



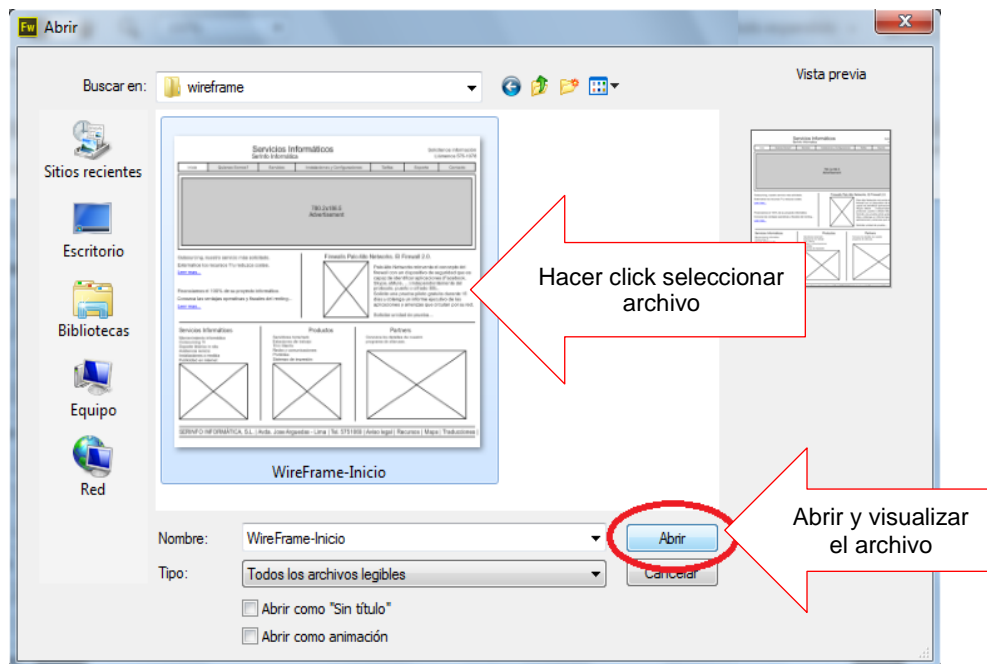
A continuación se visualiza el documento a continuación:



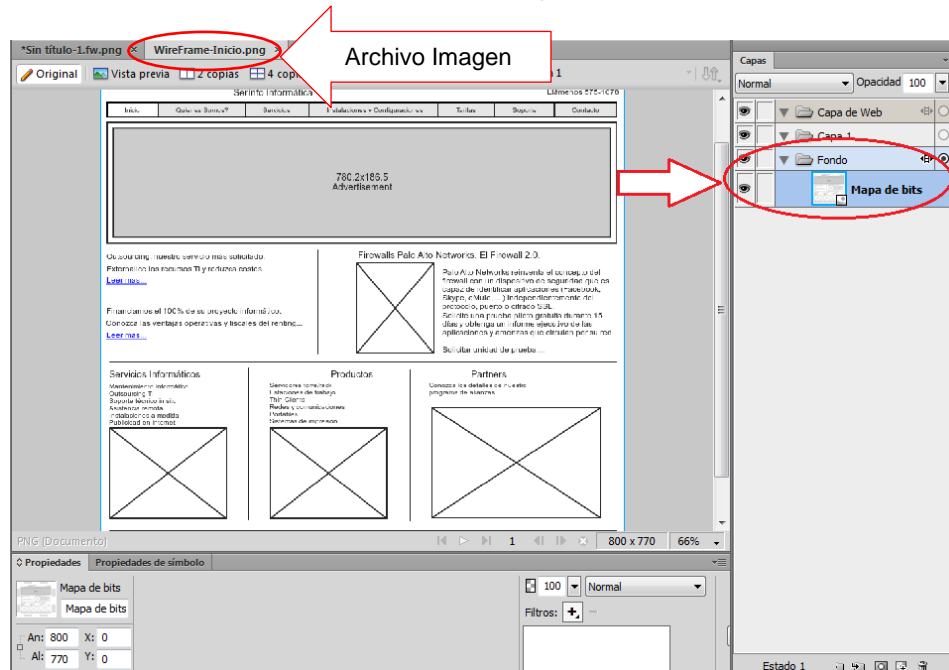
A continuación vamos a abrir el archivo wireframe. Desde la opción Archivo selecciona la opción ABRIR, tal como se muestra.



Selecciona el archivo wireFrame y presiona el boton ABRIR, tal como se muestra



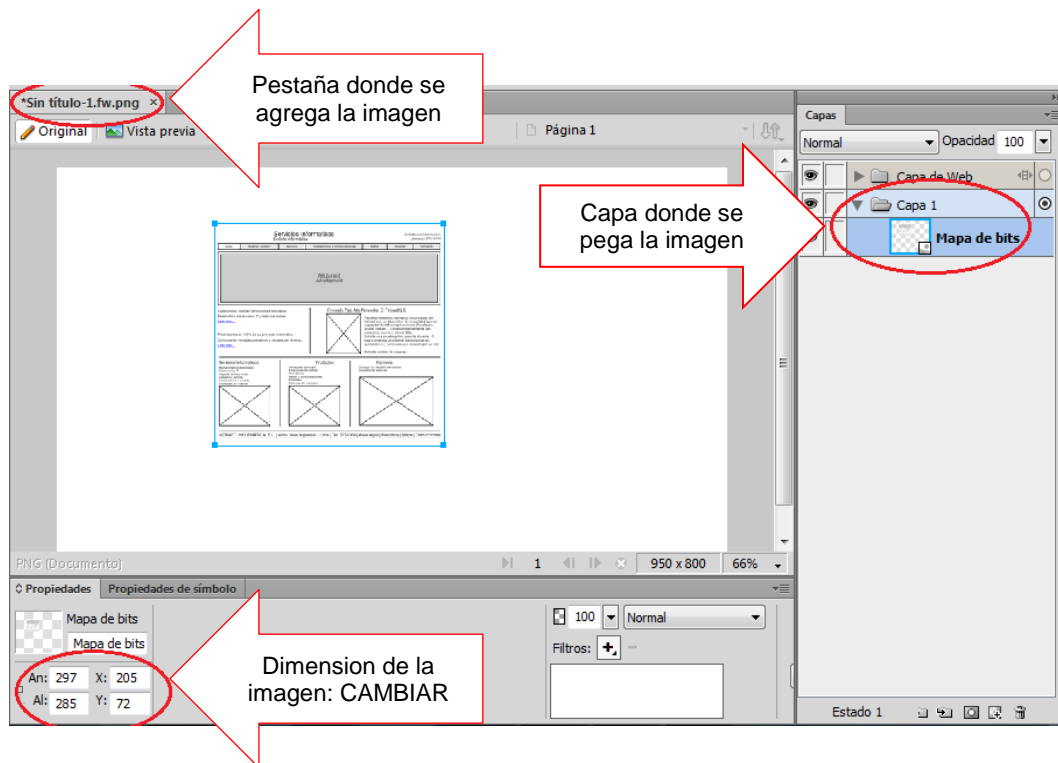
El archivo wireFrame-inicio, se visualiza en una capa "Fondo", tal como se muestra



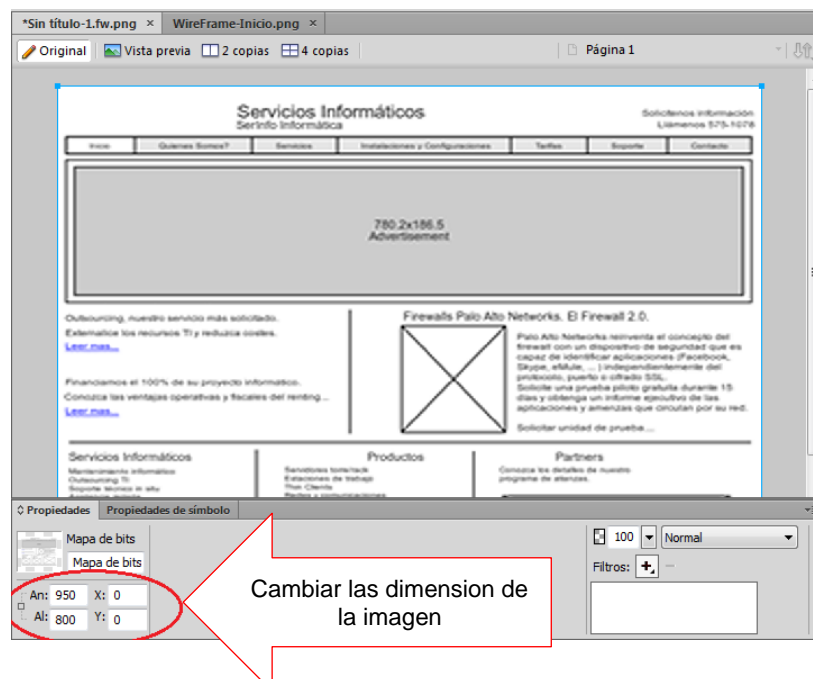
## AGREGAR EL WIREFRAME AL DOCUMENTO

A continuación vamos a copiar la imagen y pegarla en el nuevo documento:

1. Seleccionar la imagen agregada: presionamos Ctrl + A y Ctrl + C.
2. Selecciona la pestaña inicial del nuevo documento,
3. Selecciona la capa "Capa 1" y presionas Ctrl + V para pegar, nos debe quedar de esta manera

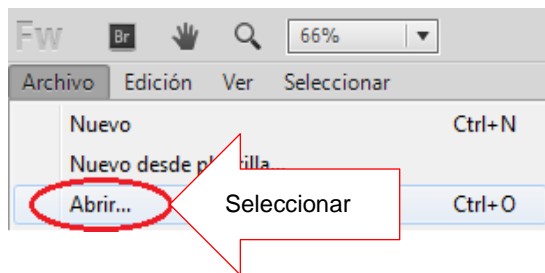


Cambiar el Ancho, Altura y las coordenadas X e Y, donde el wireframe ocupará toda la dimensión del lienzo, tal como se muestra

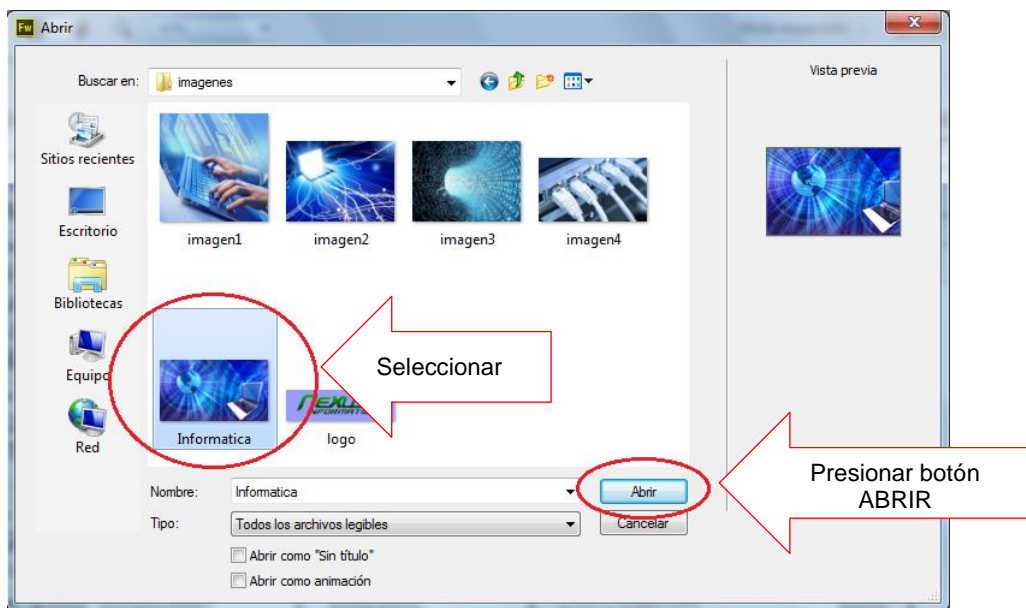


## AGREGAR UNA IMAGEN AL DOCUMENTO

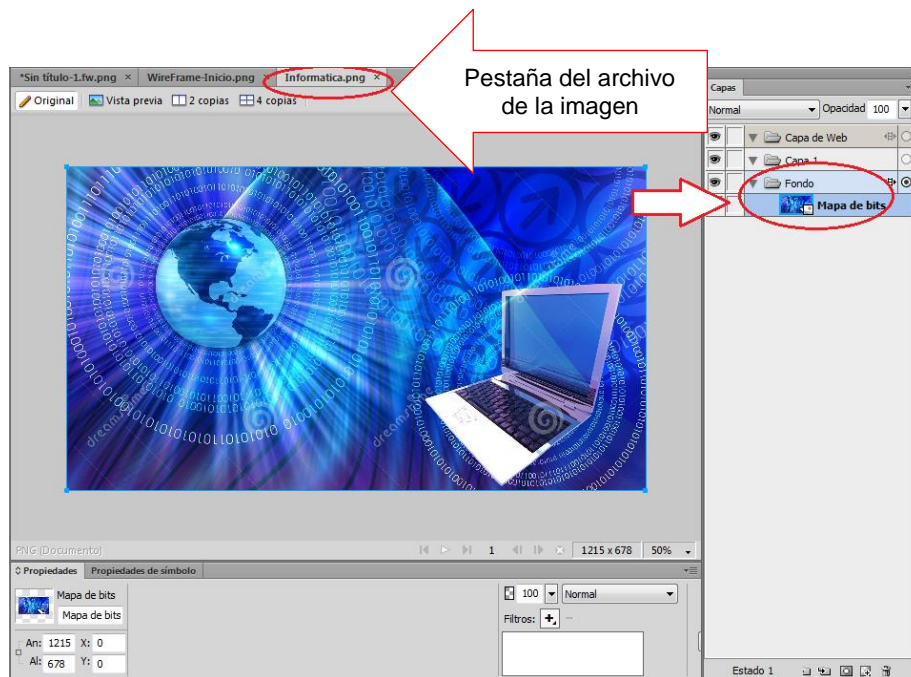
A continuación vamos a abrir una imagen para diseñar el wireframe de alta fidelidad. Desde la opción Archivo selecciona la opción ABRIR, tal como se muestra.



Selecciona el archivo de imagen, tal como se muestra y presiona el botón ABRIR

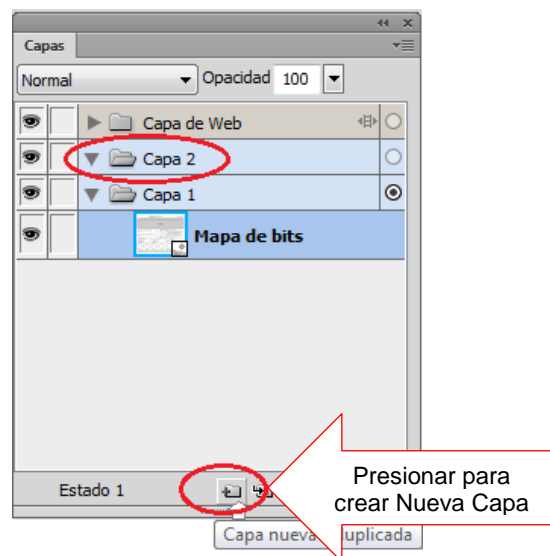


Se agrega la pestaña donde se visualiza el archivo de la imagen, tal como se muestra



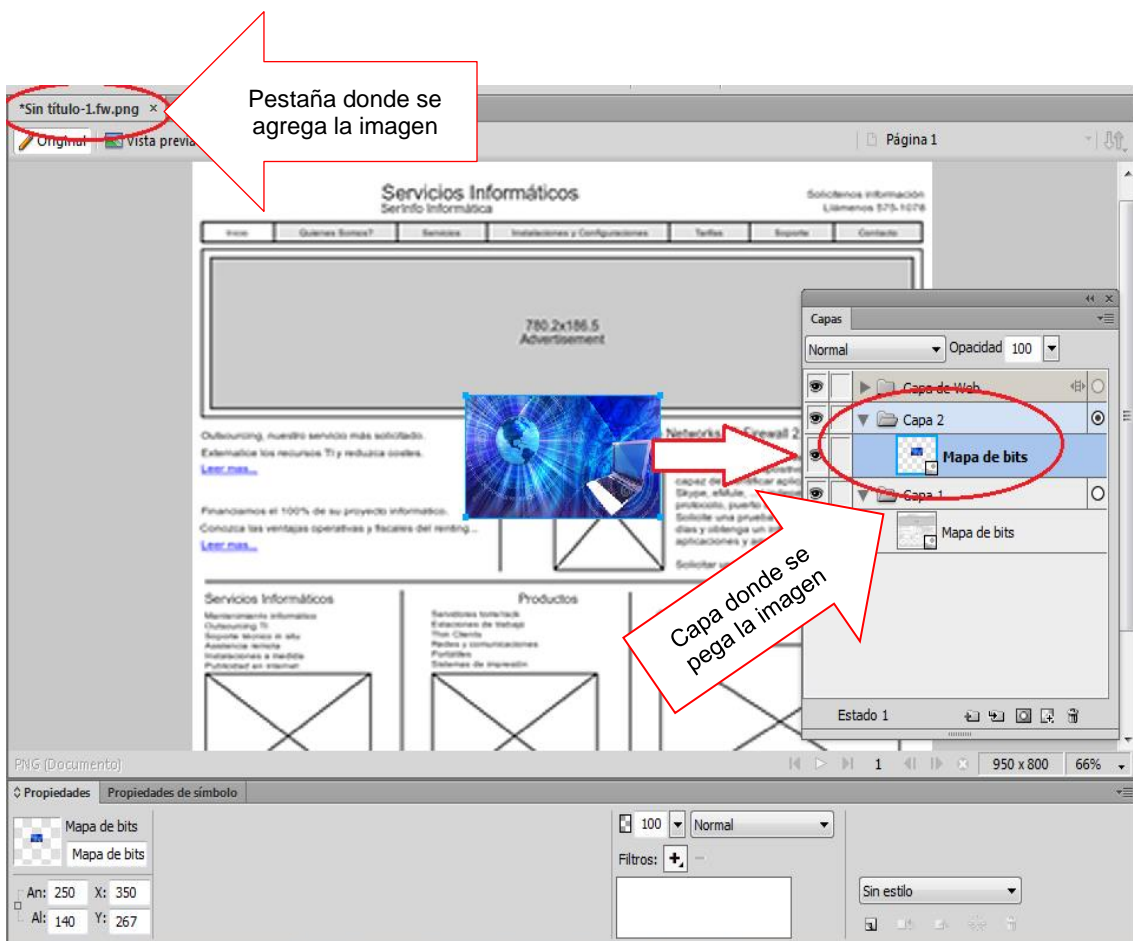
Para agregar la imagen en el nuevo documento:

1. Agregar una nueva capa: presionar el boton capa Nueva y creamos la Capa 2, tal como se muestra.



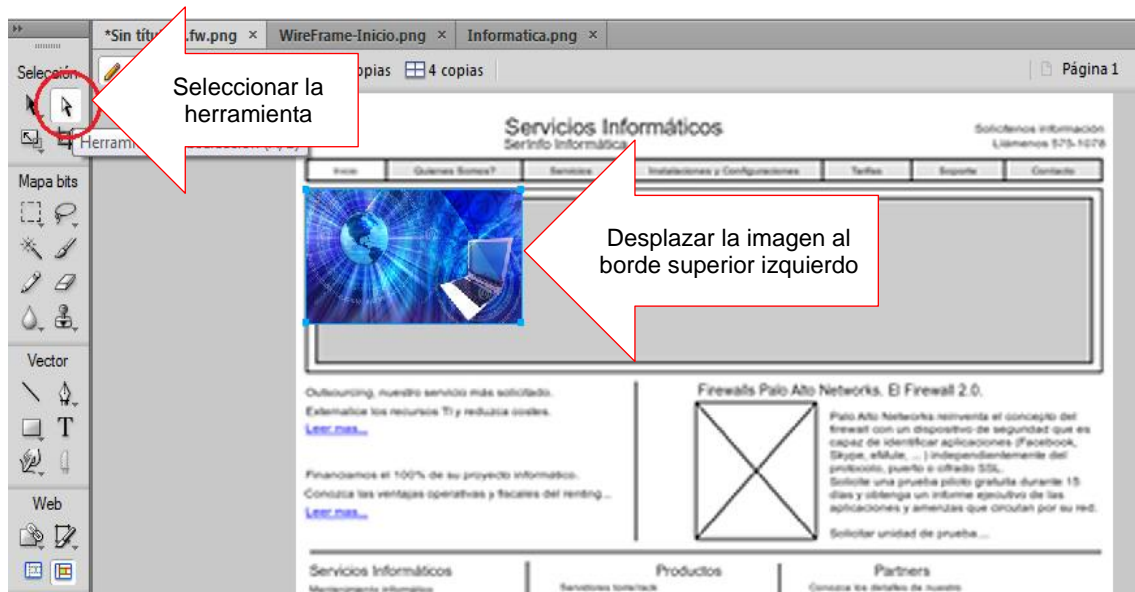
Agregada la nueva capa, vamos a copiar la imagen y pegarla en el nuevo documento:

1. Desde la pestaña de la imagen: presionamos Ctrl + A y Ctrl + C.
2. Seleccionar la pestaña del nuevo documento
3. Selecciona la nueva capa: Capa 2
4. Luego presionar Ctrl + V para pegar la imagen en la capa 2, nos debe quedar de esta manera



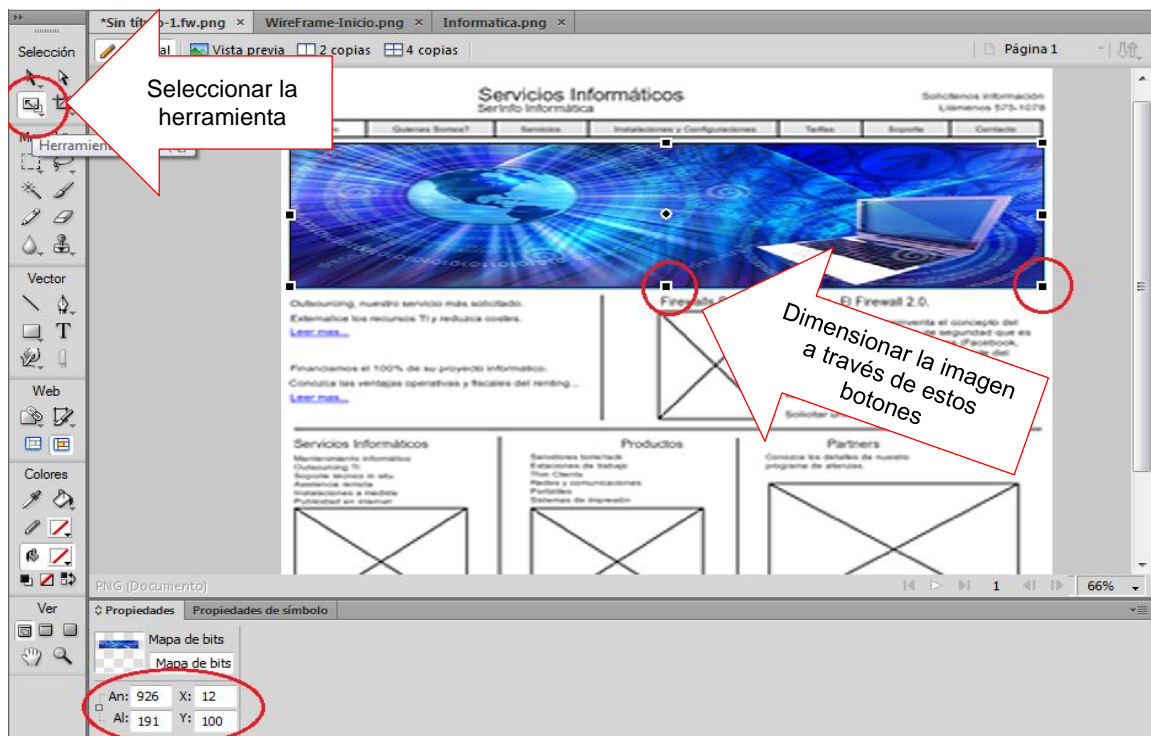


Para ubicar la imagen en la sección Advertisement del wireframe:  
 Selecciona la herramienta selección, para desplazar la imagen en el borde superior izquierdo del marco Advertisement



Para dimensionar la imagen, presiona la herramienta Escala (ver imagen), a continuación proceder a dimensionar la imagen hasta completar el tamaño del cuadro, tal como se encuentra.

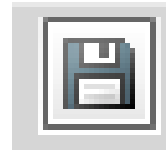
Al terminar presiona la Herramienta seleccionar (flecha)



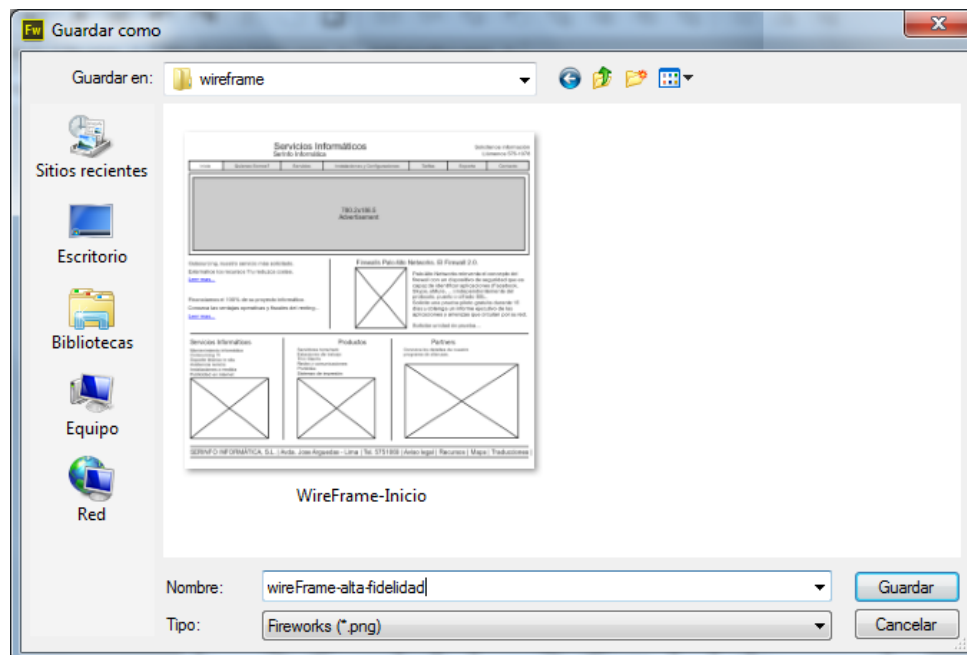


## GUARDAR EL ARCHIVO

Terminado el proceso de diseño del wireframe de alta fidelidad, debemos guardar el archivo. Presiona el botón GUARDAR.



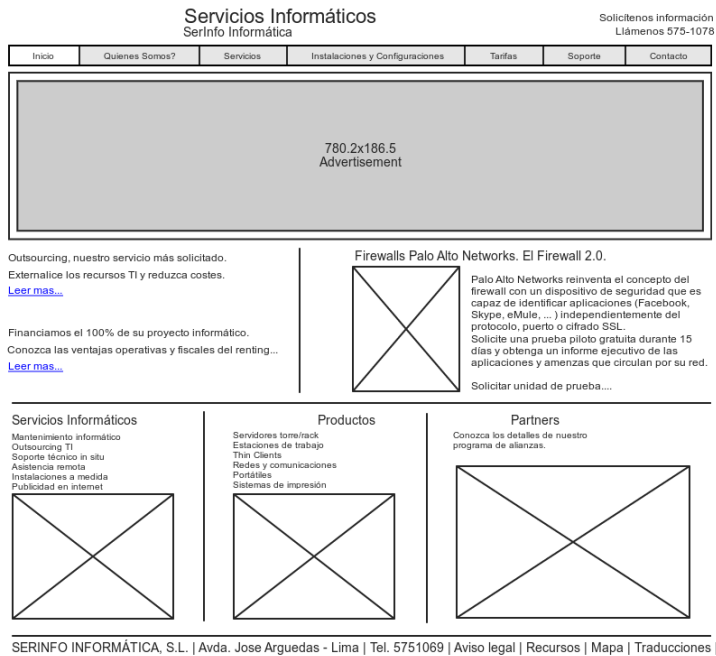
Selecciona la carpeta y asigne el nombre del archivo. Presiona el botón GUARDAR



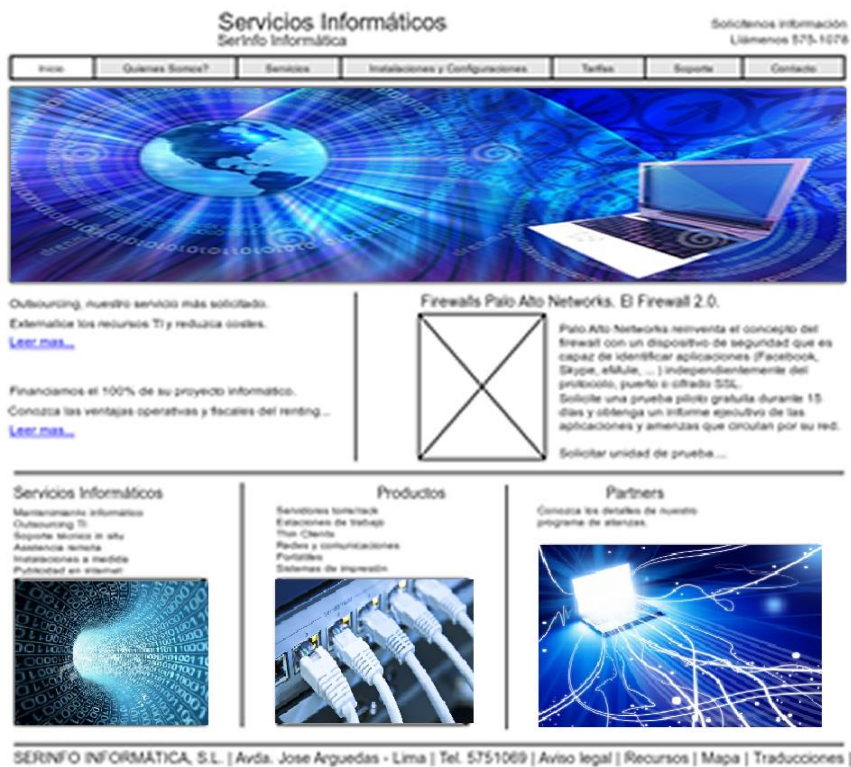
## LABORATORIO 2.2

### Trabajando con FIREWORKS

Diseña un wireframe de baja fidelidad, tal como se muestra



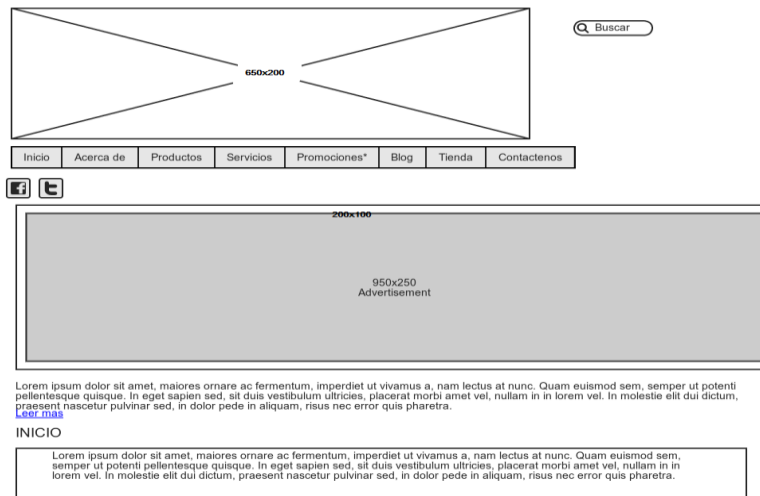
Se pide diseñar un wireframe de alta fidelidad utilizando el aplicativo FireWorks



## LABORATORIO 2.3

### Trabajando con FIREWORKS

Diseña un wireframe de baja fidelidad, tal como se muestra



Se pide diseñar un wireframe de alta fidelidad utilizando el aplicativo FireWorks



## Resumen

- 📖 Los elementos gráficos son una parte fundamental del diseño en todas sus modalidades, introduciendo en las composiciones información visual que complementa en gran medida el mensaje que se desea transmitir, y que a veces es tan importante en el diseño como los contenidos textuales del mismo.
- 📖 Los programas gráficos son de gran ayuda a la hora de trabajar con ilustraciones, ya que en ellos se realizan fácilmente operaciones que en dibujo o pintura son muy laboriosas (tareas repetitivas, copiar una imagen entera o parte de ella, dibujar trazados geométricos perfectos, aplicar filtros o efectos, etc.).
- 📖 Los editores de imágenes son aplicaciones informáticas que permiten el tratamiento y manipulación de imágenes, fotografías, vídeos, etc., de la forma más simple (reducir, ampliar, retocar, recortar...) hasta la más compleja (hacer montajes y arreglos, animar vídeos y figuras...) para mejorar su presentación
- 📖 Un editor de gráficos vectoriales es una aplicación informática que permite al usuario crear y editar imágenes de graficos vectoriales (imagen digital formada por objetos geométricos independientes, cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc.), de forma interactiva en la pantalla de la computadora y guardarlas en uno de los tantos formatos de graficos como pdf.
- 📖 Y de editors de mapa de bits es una aplicación informática que permite al usuario crear y editar imágenes de mapa de bits (imagen que esta formada por pixeles).
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 <http://www.desarrolloweb.com/articulos/2063.php>

🔗 <http://www.desarrollomultimedia.es/teoria-del-diseno/manual-photoshop.html>

🔗 <http://support.microsoft.com/kb/307205/es>

🔗 <http://msdn.microsoft.com/es-es/library/ms173152.aspx>



# HTML5 Y CSS3

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3.

## TEMARIO

### 2.1 Tema 3: Estructura HTML y aplicación de estilos CSS

#### 2.1.1 Entorno del programa (Editor HTML)

#### 2.1.2 Creación de un sitio web (estructura de carpetas)

#### 2.1.3 Estructura de un documento HTML

##### 2.1.3.1 Estructura del documento

##### 2.1.3.2 Etiquetas básicas: encabezados (H1...H6), párrafos, listas, enlaces internos y externos, imagen.

### 2.2 Tema 4: Estilos CSS

#### 2.2.1 Estilos CSS

##### 2.2.1.1 Tipos de Estilos: En línea e internos.

##### 2.2.1.2 Estilos aplicados a etiquetas y de clase.

## ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5 y hojas de estilo CSS3.
- Los alumnos diseñan páginas web utilizando bloques de contenidos.



## 1.4 ESTRUCTURA HTML Y APLICACIÓN DE ESTILOS CSS

Desde hace más de dos décadas HTML ejerce una función primordial en el proceso de publicar información a través de la Internet.

Ante las necesidades de los usuarios y productores de contenido se ha lanzado la versión HTML5. Esta nueva versión del lenguaje Web simplifica el trabajo y agrupa las nuevas tecnologías de desarrollo de aplicaciones web: CSS3 y Javascript.



Entre los principales cambios del HTML5, tenemos

- Cambios en el etiquetado, elementos más semánticos, formularios más potentes.
- Cambios en el estilo, nuevos selectores, uso de tipografías, columnas, rejillas, y mejoras visuales
- Mejor soporte multimedia, soporte de video y audio, canvas, WebGL, SVG, etc.
- Interactividad, geolocalización, almacenamiento local, sockets, acceso a hardware, etc.
- Mejor Javascript, una API mejorada para acceder a los elementos de un modo más eficiente
- Nuevo estilo, mejor CSS tenemos nuevos selectores que permiten seleccionar mejor los elementos HTML

Para construir una página web, no necesitas más que un editor de texto básico como Bloc de Nota o Notepad (Windows), o Text Edit (Mac), pero no son programas mejor preparados para trabajar con páginas web.

Existen multitud de programas gratuitos para editar páginas web. Aquí tiene algunos:

- **WebFacil**: editor gratuito y sencillo para editar páginas web, se encuentra en castellano
- **NVU**, es un editor gráfico HTML, similar al Front Page, está basado en el código de Mozilla Compose y que permite crear contenidos web como si trabajásemos con un procesador de textos.
- **CoffeCup Free Editor**, es un editor multiplataforma (Windows, Mac) y gratuito con soporte para multitud de características: ayuda contextual a la escritura de código, vista previa del resultado a medida que escribimos, etc.
- **SublimeText**, Un editor de texto avanzado para programadores, multiplataforma. Aunque es bastante espartano lo cierto es que todo el que invierte tiempo en aprender a dominarlo luego no lo quiere dejar, ya que aumenta mucho la productividad a la hora de escribir código. No ofrece muchas de las características de ayuda al programador web que sí ofrecen

la mayoría de los otros entornos, por lo que es recomendable sobre todo si vamos a trabajar mucho directamente con código, especialmente JavaScript. Lo podemos descargar y utilizar gratuitamente, pero si lo vamos a usar a menudo para trabajar sus creadores nos solicitan que les paguemos una licencia (es lo justo).

- **Komposer**, Archiconocido editor de HTML y CSS que es gratuito y multiplataforma (Windows Mac y Linux). Ofrece vista previa de la página mientras se escribe código, múltiples pestañas, edición específica de CSS, etc.

Existen programas comerciales, muy potentes que disponen de un entorno visual, y generan automáticamente el código de las páginas. Al poder ver en todo momento como quedara la página en el navegador, se facilita la creación de las páginas, y el uso de menus permite ganar rapidez.

**Dreamweaver** es el mejor editor para crear páginas web, permite crear avanzadas web que se pueden visualizar también en dispositivos móviles, sin tener que dominar el código html, combina las últimas tecnologías en desarrollo web como Javascript, CSS, AJAX, XHTML, Adobe AIR, Smart Objects de Photoshop, subversiones (SVN), frameworks Javascript y más. Es compatible con Google Chrome, Internet Explorer, Opera, Safari, Firefox y el resto de navegadores, además soporta CMS como WordPress, Joomla! y Drupal.

Para asegurar que pueda experimentar todos estos nuevos recursos, se recomienda que instale en su sistema las últimas versiones de los siguientes navegadores Web, cuando esté desarrollando aplicaciones HTML5 y CSS3:

- Mozilla Firefox (versión 3.5+)
- Apple Safari (versión 4.0+)
- Opera (versión 10.0+)
- Google Chrome (versión 3.0+)

### 1.6.1 Entorno del programa

Dreamweaver tiende a ser el software de diseño de sitios web elegido por muchos diseñadores con experiencia moderada y también por expertos. Es mucho más que un editor de HTML, Dreamweaver logra integrar completamente el diseño visual con las herramientas de codificación. También es compatible con una variedad de diferentes lenguajes de scripting, incluyendo PHP, ASP y CSS.





Dreamweaver proporciona a los usuarios un alto nivel de control a lo largo de todo el proceso de diseño. El nivel de personalización que ofrece el software es prácticamente inigualable. Dreamweaver te permite crear funciones interactivas avanzadas, como menús desplegables, imágenes de sustitución, diseños de la red de fluidos, aplicaciones sencillas para dispositivos móviles y paneles plegables.

Gracias a la avanzada capacidad de WYSIWYG, la creación de un sitio web sencillo con Dreamweaver es, en realidad, bastante fácil. Dreamweaver desarrollará código HTML por ti; también otros programas de software WYSIWYG lo harán por ti, por lo que no tienes que ser un genio de la codificación para crear un sitio web. El Design View de Dreamweaver te proporcionará el editor de WYSIWYG más avanzado disponible en el mercado hoy.

Debido a que Dreamweaver ofrece edición híbrida, si tienes conocimientos de codificación, puedes trabajar en un entorno de doble panel y así aprovechar la codificación de WYSIWYG y la codificación manual al mismo tiempo. Y si ya eres un experto en codificación, puedes trabajar exclusivamente en código, sólo haciendo el cambio a modo visual para ver el producto final. Para terminar, Dreamweaver es compatible con docenas de plug-ins, por lo que el diseño web resulta aún más fácil.



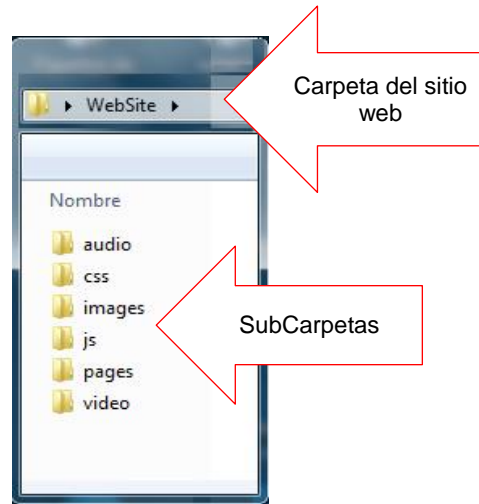
Para un programador que inicia aprender a diseñar páginas web con Dreamweaver, debes tener en cuenta que es un programa muy complejo, sobre todo si apenas estás empezando con el diseño de sitios web. Te convendría tomar un curso acerca de Dreamweaver. También puedes instruirte en otras herramientas de diseño de Adobe en este curso acerca de Adobe Muse.

### 1.6.2 Creación de un sitio web

Un sitio Web ofrecerá tanto a clientes como a proveedores y empleados, un canal de comunicación abierto las 24 hrs. del día con información completa y actualizada sobre la organización, para que realicen consultas, pedidos o sugerencias.

### Creando un sitio web en Dreamweaver

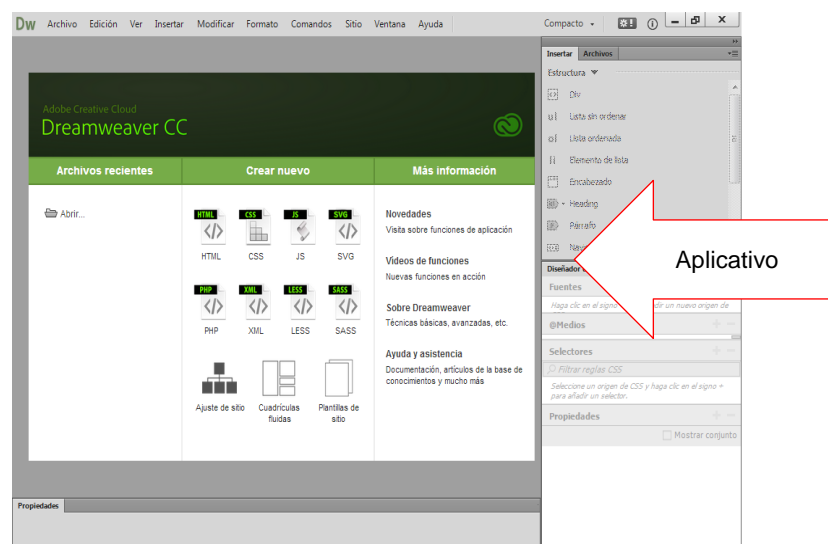
Para que Dreamweaver le ayude eficazmente en la confección de un sitio web, se debe seguir una serie de pasos, como comunicarle a Dreamweaver dónde se encuentran en su ordenador los componentes del sitio, por ejemplo, hojas de estilo CSS, imágenes o documentos HTML.



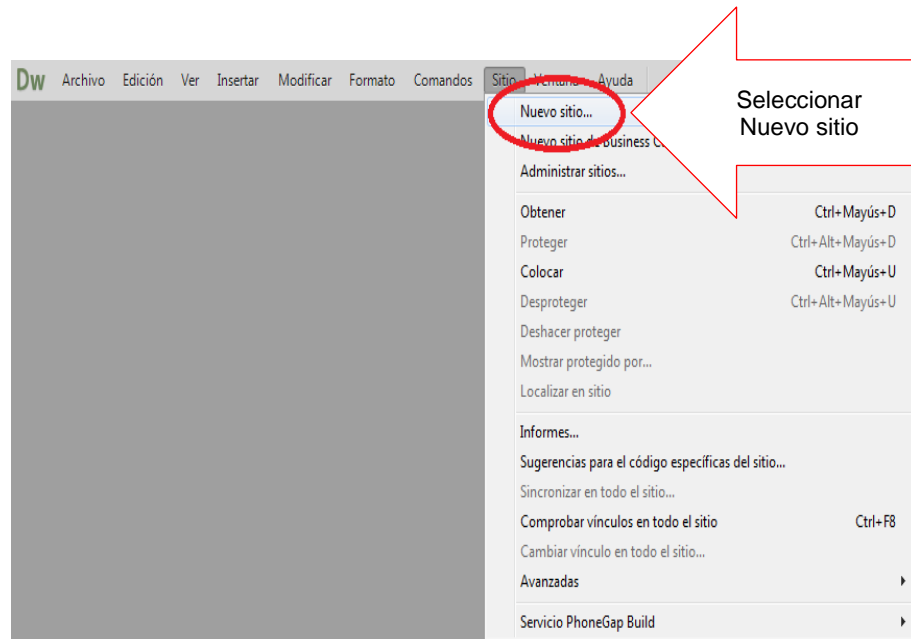
Estos archivos deben estar organizados en subcarpetas dentro de una carpeta principal. Con una estructura ordenada, Dreamweaver podrá controlar los cambios que se produzcan en los archivos y mantendrá actualizados los enlaces de su sitio web. Sin esta organización de datos a través de Dreamweaver, el cambio de una sola imagen podría provocar una serie de errores en su sitio web.

Para crear un sitio web por Dreamweaver

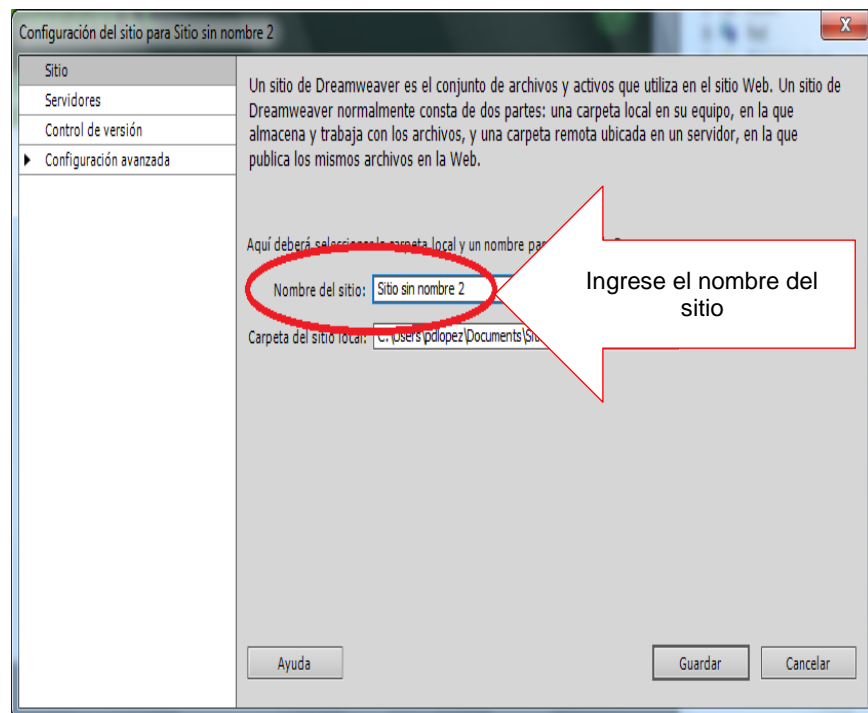
#### 1. Abrir el aplicativo Dreamweaver



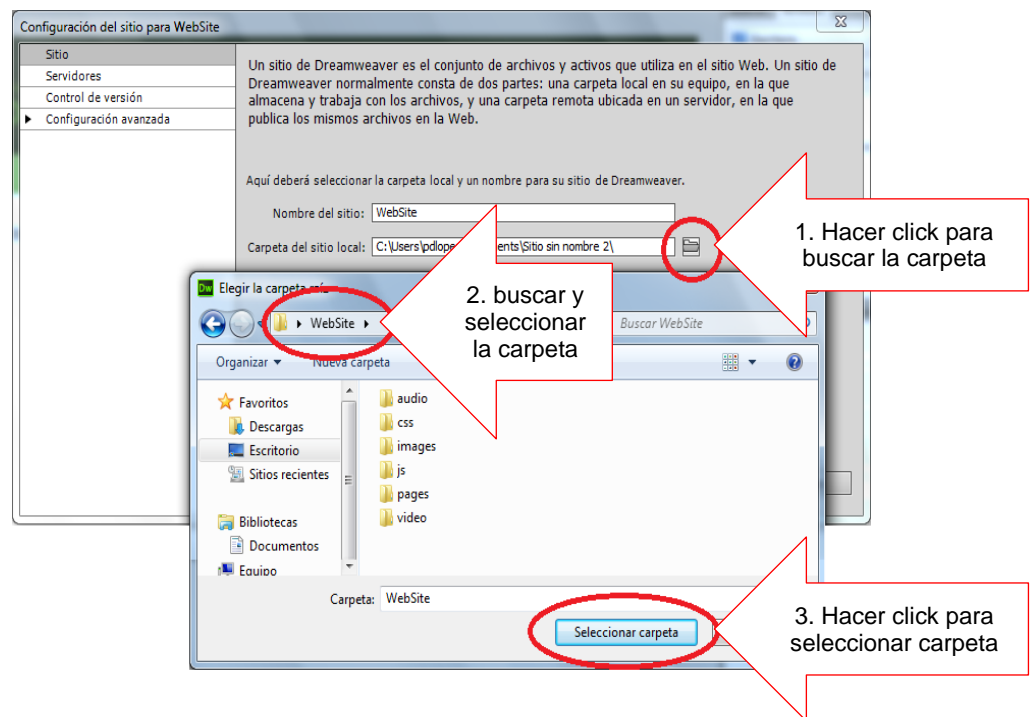
2. Desde la opción Sitio del menú principal selecciona Nuevo sitio



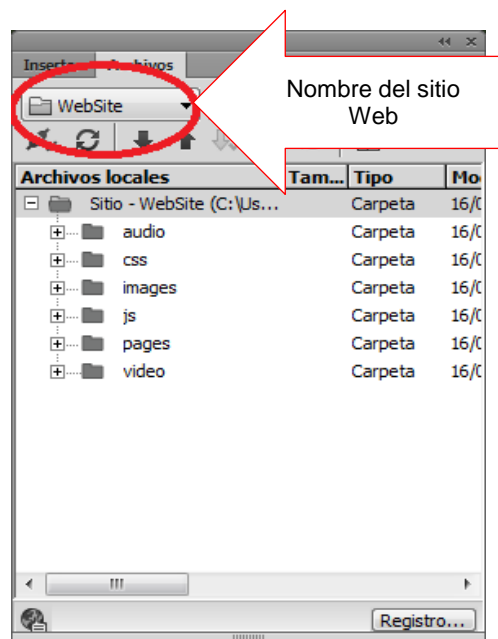
3. Ingrese el nombre del sitio Web, tal como se muestra



4. Seleccionar la carpeta del sitio web Local, tal como se muestra, luego presiona el botón GUARDAR

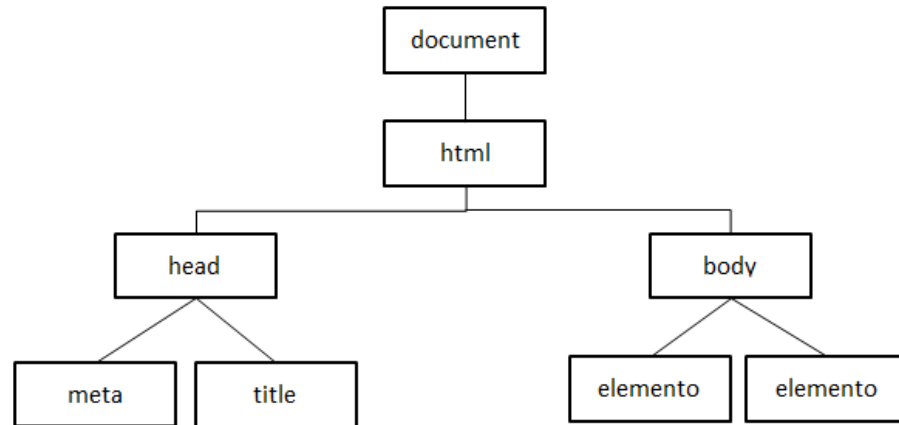


5. Al finalizar el sitio web se visualiza en la ventana "Archivos", donde aparece el nombre del Sitio Web y su lista de sub carpetas, tal como se muestra.



### 1.6.3 Estructura de un documento HTML

Un documento HTML no es más que un archivo de texto. La extensión de su nombre suele ser .html o .htm.



Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto, imágenes, videos, audio, etc.

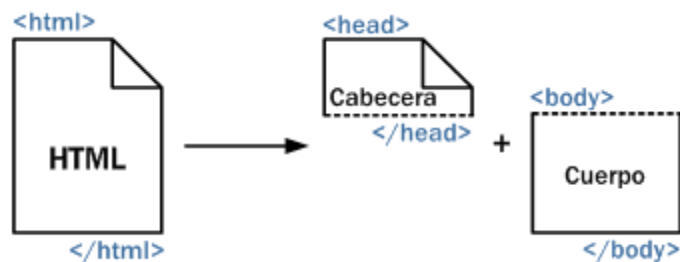


Figura 1

Referencia: [http://librosweb.es/xhtml/capitulo\\_2/el\\_primer\\_documento\\_html.html](http://librosweb.es/xhtml/capitulo_2/el_primer_documento_html.html)

#### 2.1.3.1 Elementos de una página html

##### Elemento doctype

La estructura de un documento en HTML5 inicia con la declaración del tipo de documento que vamos a utilizar, esto lo hacemos definiendo la etiqueta **doctype** de la siguiente forma:

```
<!doctype html>
```

##### Elemento html

A continuación comenzamos a construir la estructura del documento HTML definiendo las etiquetas `<html>` y `</html>`.

Esta etiqueta delimita el documento HTML, indicando al navegador el comienzo y fin de la página html. En esta etiqueta vamos utilizar el atributo **lang** que nos servirá para especificar el idioma del documento, en este caso español por tanto le asignamos el valor **es**.

```
<!doctype html>
<html lang="es">

</html>
```

### Elemento head

La palabra **head** viene del inglés “cabeza” y su función es delimitar cabecera del documento. Sus etiquetas son: **<head>** y **</head>**.

La cabecera es la sección apropiada para incluir información sobre el documento, la mayoría de la cual no será mostrada a los lectores. Para incluir esta información tenemos diversos elementos, de momento solo comentaremos el más importante:

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Untitled Document</title>
</head>
<body>

</body>
</html>
```

La directiva **<title>** sirve para especificar el título del documento y es visualizada en la barra de título de la ventana correspondiente al programa navegador.

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Página Principal</title>
</head>
<body>

</body>
</html>
```

La directiva **<meta>** es utilizada en el documento, donde indicamos (a través del atributo **charset**) al navegador que tipo de codificado es el documento html, de preferencia usamos el **UTF-8**.

Utilizando la etiqueta **<meta>** podemos describir el documento, a través del atributo **name** la palabra clave **description** indica que es la descripción del documento y en el atributo **content** el texto que conforma la descripción del documento.

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="Mi pagina">
    <title>Pagina Principal</title>
</head>
<body>

</body>
</html>
```

Continuando con las etiquetas **<meta>** definiremos las palabras clave **keywords** que describen el contenido de nuestro documento esto para ayudar a los buscadores a identificar sobre que trata nuestro documento HTML.

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="Mi pagina">
    <meta name="keywords" content="HTML,CSS">
    <title>Pagina Principal</title>
</head>
<body>

</body>
</html>
```

La directiva **<link>** es usado para incorporar estilos, códigos Javascript, imágenes o iconos desde archivos externos. Uno de los usos más comunes para **<link>** es la incorporación de archivos con estilos CSS:

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Pagina Principal</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
</body>
</html>
```

## Elemento body

La etiqueta **<body>** delimita el cuerpo del documento HTML. Este elemento contiene todo el contenido visible de un documento HTML (párrafos de texto, imágenes, tablas). La etiqueta **<body>** es compatible con todos los navegadores.

### 2.3.1.2 Etiquetas básicas: encabezados (H1...H6), párrafos, listas, enlaces internos y externos, imagen.

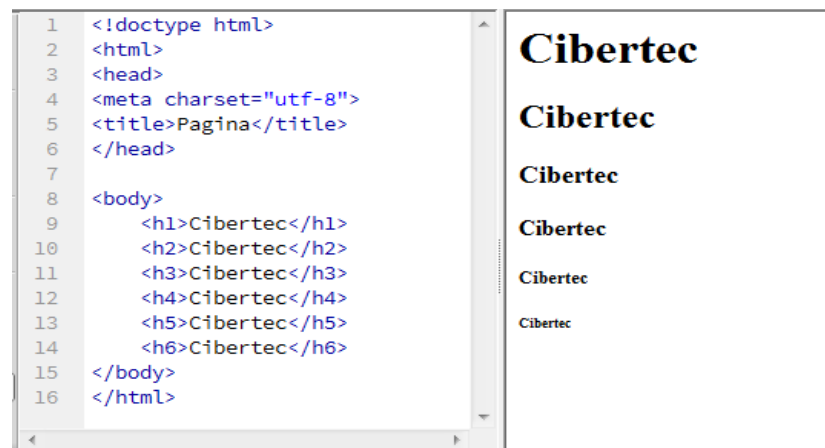
#### Etiquetas Headings

Nos definen el tamaño de un título o cabecera

**<h1>** nos dá el tipo de letra más grande

**<h6>** nos dá el tipo de letra más pequeño.

HTML agrega automáticamente un espacio antes y después de cada título. **<h1>** al ser usado como título de una página web, es de suma importancia ya que es uno de los parámetros de Google y demás buscadores tiene en cuenta a la hora de indexa un sitio web.



#### Etiqueta párrafo

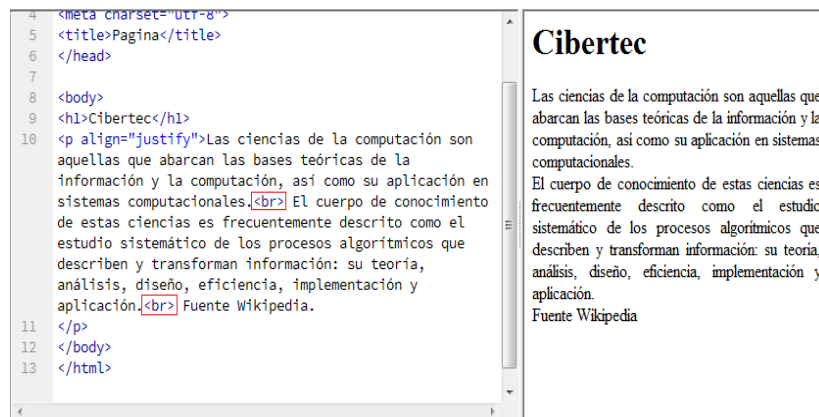
La etiqueta párrafo se define con la etiqueta **<p>**. Esta etiqueta permite definir los párrafos que forma el texto de una página. El texto del párrafo lo puedes alinear utilizando la etiqueta **align** utilizando los parámetros “center”, “right”, “left” y “justify”





### Etiqueta salto de línea

La etiqueta **<br>** incluye una nueva línea y fuerza a que el texto se muestre en la siguiente línea. Es el equivalente a presionar la tecla ENTER (o Intro).

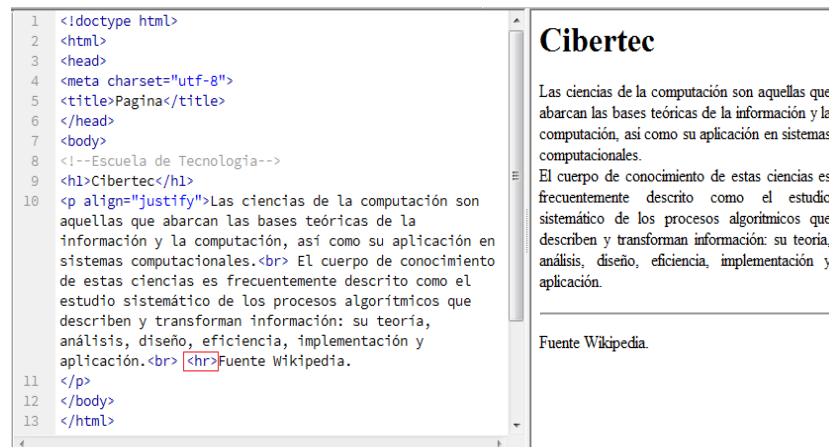


### Etiqueta trazo de línea

La etiqueta **<hr>** nos permite trazar una línea horizontal.

### Etiqueta comentario

La etiqueta **<!-- ... -->** se utiliza para insertar un comentario dentro del código que estamos escribiendo. El mismo es ignorado por el navegador al momento de leerlo. Los comentarios nos sirven para explicar mejor el código y son de gran ayuda en el momento que necesitemos editarlo.



### Etiquetas de marcado

Una vez estructurado el texto en párrafos y secciones, el siguiente paso es el marcado de los elementos que componen el texto. Los textos habituales están formados por elementos como palabras en negrita o cursiva, anotaciones y correcciones, citas a otros documentos externos, etc. HTML proporciona varias etiquetas para marcar cada uno de los diferentes tipos de texto.

Etiqueta	Descripción
<b>&lt;em&gt;</b>	Marca un texto indicando que su importancia es mayor que la del resto del texto
<b>&lt;strong&gt;</b>	Indica que un determinado texto es de la mayor importancia dentro de la página

<blockquote>	Se emplea para indicar que el texto que encierra es una cita textual de otro texto externo
<ins>	Define un texto subrayado
<sub>	Define un texto subíndice
<sup>	Define un texto superíndice
<small>	Define un texto pequeño
<big>	Define un texto grande
<del>	Define un texto tachado
<i>	Define el texto en itálica
<b>	Define el texto en negrita

### Etiquetas de Enlace

Los enlaces o *links* (en inglés) nos permite conectar con otros documentos:

- Una imagen
- Un video
- Un archivo de sonido
- Una pagina de un determinado sitio web
- Enviar un email

Los enlaces en HTML se crean mediante la etiqueta **<a>** (viene del inglés “anchor” traducido literalmente sería “ancla”).

El atributo más importante de la etiqueta **<a>** es **href**, que se utiliza para indicar que apunta a ese enlace. Las URL de los enlaces pueden ser: absolutas, relativas, internas y externas.

Un **Enlace externo** apunta a páginas de otros sitios web; es decir, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Un **Enlace interno** apunta a páginas del propio sitio web. Las **URL absolutas** incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado. Las **URL relativas** se construyen a partir de la URL absolutam donde prescindimo de la parte del protocolo, del nombre del servidor e incluso de parte o toda la ruta del recurso enlazado.

Para crear un enlace que apunte a la página principal de Cibertec solamente habría que incluir lo siguiente en un documento HTML:

```
<a href="http://www.cibertec.edu.pe">Cibertec</a>
```

Otro atributo de la etiqueta **<a>** es **name**, permite definir enlaces dentro de una misma página web. Es útil cuando la página es muy larga, donde se puede mostrar enlaces como: "Ir hasta la segunda sección", "Volver al principio de la página", etc.

```
<a name="ir_inicio"></a>
```

La sintaxis que se utiliza con estos enlaces es la misma que con los enlaces normales, salvo que se añade el **símbolo #** seguido del nombre de la sección a la que se apunta.

```
<a href="#ir_inicio">Ir al Inicio</a>
```

Los enlaces directos a secciones también funcionan con el **atributo id** de cualquier elemento.

```
<h1 id="inicio">Título de la página</h1>

<a href="#inicio">Volver al inicio de la página</a>
```

El atributo type se utiliza para notificar al navegador sobre el tipo de contenido que se enlaza. Se indica mediante una cadena de texto cuyos posibles valores también están estandarizados. Los valores de los contenidos más utilizados son los siguientes:

- "text/html" (páginas HTML),
- "image/png" (imágenes con formato PNG),
- "image/gif" (imágenes con formato GIF),
- "text/css" (hojas de estilo CSS),
- "application/rss+xml" (archivos RSS).

```
<a href="foto.jpg" type="image/jpg">Imagen Principal</a>
```

Para definir un enlace desde cualquier página web, se vuelve directamente a la página de inicio o página principal del sitio web.

```
<a href="/">Inicio</a>
```

A través del valor **mailto**, se abre automáticamente el programa de correo electrónico del ordenador del usuario y se establece la dirección de envío al valor indicado después de **mailto**:

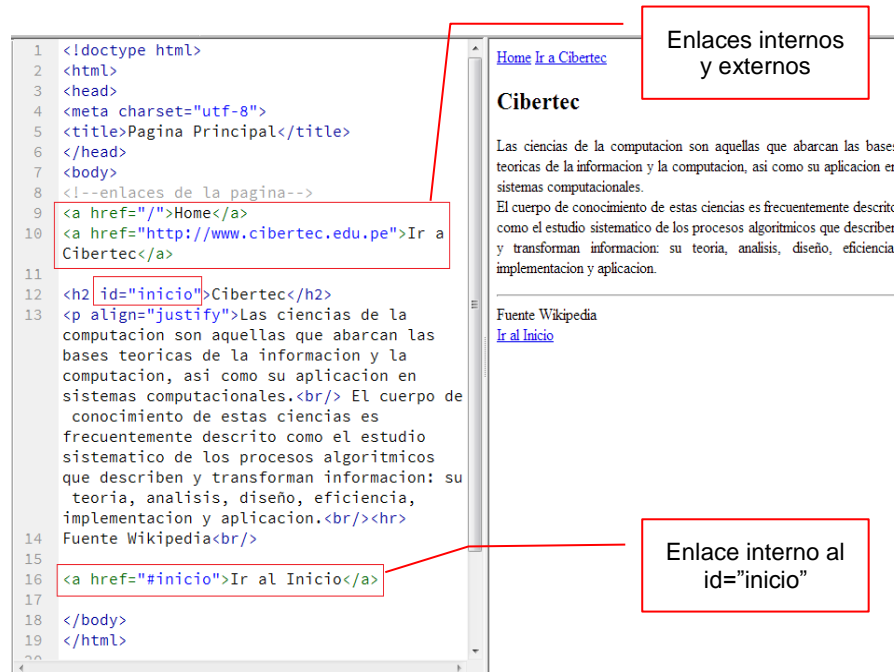
```
<a href="mailto:nombre@direccion.com" title="Email para sol
licitar más información">Solicita más información</a>
```

Aunque el uso de **mailto** puede parecer una ventaja, su uso está desaconsejado. Si incluimos nuestro email directamente en una página web, en poco tiempo ese email estará lleno de correo electrónico basura o "spam", ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

El atributo **target** se utiliza para definir donde queremos que se abra el documento al presionar sobre el enlace, utiliza los siguientes valores:

- **\_blank**: Indica una nueva ventana.
- **\_parent**: Especifica el marco matriz que contiene el vínculo de origen.
- **\_self**: El significado es muy evidente. El nuevo documento se carga en el mismo marco.
- **\_top**: Esto indica una nueva ventana de documento.

Acontinuación presentamos un ejemplo utilizando enlaces.



## Etiquetas de Imagen

Las imágenes de contenido son las que proporcionan información y complementan la información textual. Las imágenes de adorno son las que se utilizan para hacer bordes redondeados, para mostrar pequeños iconos en las listas de elementos, para mostrar fondos de página, etc.

Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta **<img>** y las imágenes de adorno no se deberían incluir en el código HTML, sino que deberían emplearse hojas de estilos CSS para mostrarlas.

Los dos atributos requeridos son **src** y **alt**. El atributo **src** representa la dirección URL de la imagen; la URL pueden ser absoluta o relativa. El atributo **alt** permite describir el contenido de la imagen mediante un texto breve. Las descripciones deben tener una longitud inferior a 1024 caracteres y son útiles para las personas y dispositivos discapacitados que no pueden acceder a las imágenes.

```

```

Los atributos **width** y **height** se utilizan para indicar la anchura y altura con la que se muestran las imágenes. Como ya se ha comentado, HTML estructura de forma correcta los contenidos de la página y CSS define el aspecto gráfico con el que se muestran los contenidos.

```

```

Si el valor del atributo **width** o **height** se indica mediante un número entero, el navegador supone que hace referencia a la unidad de medida píxel; también es posible indicar la anchura y altura en forma de porcentaje. En este caso, el porcentaje hace referencia a la altura/anchura del elemento en el que está contenida la imagen.

El atributo **border** permite asignar un border a la imagen, si su valor es cero la imagen no tendrá borde.

A continuación presentamos un ejemplo utilizando imágenes.

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Pagina Principal</title>
6 </head>
7 <body>
8 <a href="/">Home</a>
9 <a href="http://www.cibertec.edu.pe">Ir a Cibertec</a>
10
11 <h2 id="inicio">Cibertec</h2>
12 <p align="justify">
13 Las ciencias de la computacion son aquellas que abarcan
14 las bases teoricas de la informacion y la computacion, asi
15 como su aplicacion en sistemas computacionales.
16 <p>
17 
18 <p align="justify">
19 El cuerpo de conocimiento de estas ciencias es
20 frecuentemente descrito como el estudio sistematico de los
21 procesos algoritmicos que describen y transforman
22 informacion: su teoria, analisis, diseño, eficiencia,
23 implementacion y aplicacion.
24 <p>
25 
27 <p><hr>
28 <a href="#inicio">Ir al Inicio</a>
29 </body>
30 </html>

```

### Etiquetas figure

Las etiquetas `<figure>` y `<figcaption>` permiten definir una mejor semántica (significado o sentido) a nuestro contenido dentro de un documento HTML5.

En el caso de estos elementos en específico nos permitirán anotar o resaltar ilustraciones, diagramas o fotos que son referenciados directamente desde el contenido principal. Estas etiquetas podríamos utilizarlas en Diagramas, imágenes, fotos que acompañan el texto principal de nuestro contenido.

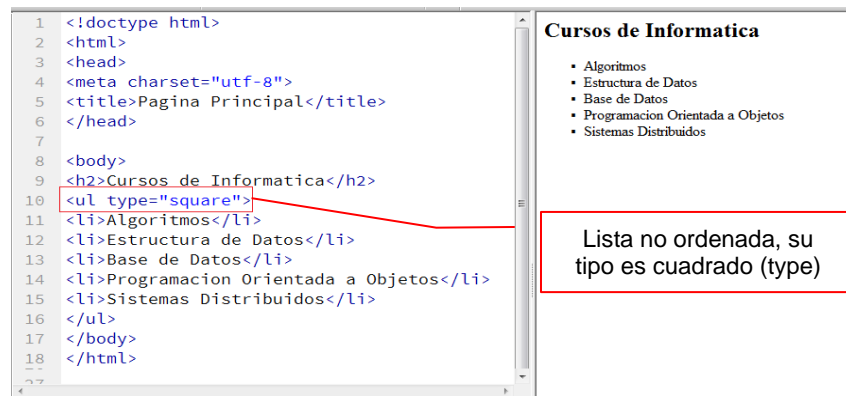
## Etiquetas de Lista

Las listas nos permiten ordenar o enumerar datos o elementos para facilitar su lectura. Tenemos tres tipos de listas:

- No ordenadas.
- Ordenadas.
- De definición.

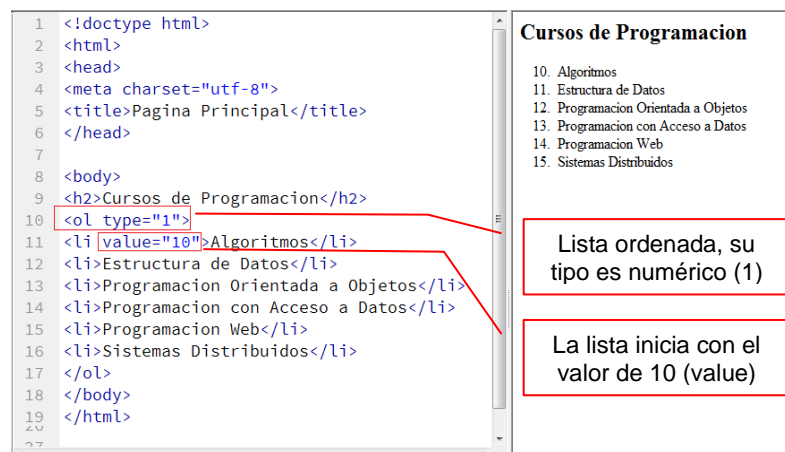
Listas no ordenadas. Son las más sencillas y las que más se utilizan. Una lista no ordenada es un conjunto de elementos relacionados entre sí, pero no se indica un orden o secuencia determinados. La etiqueta **<ul>** encierra todos los elementos de la lista y la etiqueta **<li>** representa cada uno de sus elementos.

Para definir un determinado símbolo, por ejemplo: cuadrados, se define a través del atributo **type**. Los símbolos para las listas son: disc (puntos negros), circle (círculos), square (cuadrado). A continuación presentamos un ejemplo utilizando listas no ordenadas



Listas ordenadas. Son casi idénticas a las listas no ordenadas, con la diferencia en que sus elementos se muestran siguiendo un orden determinado. La etiqueta **<ol>** encierra todos los elementos de la lista y la etiqueta **<li>** representa cada uno de sus elementos.

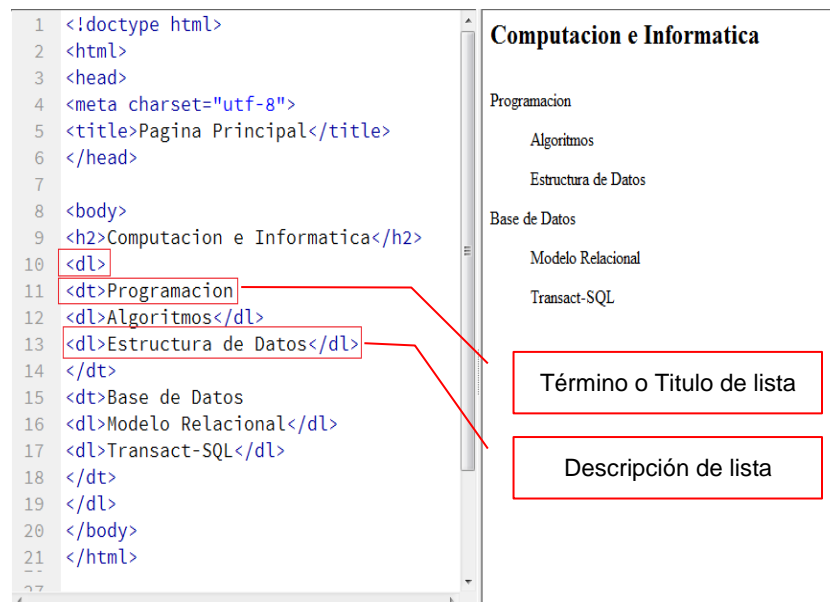
En las listas ordenadas podemos hacer que el primer punto comience con un determinado número, esto se logra a través del atributo **value**. A continuación presentamos un ejemplo utilizando listas ordenadas.



Listas de definición. Son listas que se utilizan en la mayoría de páginas HTML. Su funcionamiento es similar al de un diccionario, ya que cada elemento de la lista está formado por términos y definiciones.

La etiqueta **<dl>** crea la lista de definición y las etiquetas **<dt>** y **<dd>** definen respectivamente el término (título de la definición) y la etiqueta **<dd>** representa la descripción de cada elemento de la lista.

A continuación presentamos un ejemplo utilizando listas de definición



### Etiqueta table

Las tablas son herramientas muy útiles para presentar datos en una estructura de filas y columnas, para el diseño de páginas y ubicación de textos y gráficos dentro de las mismas.

Una tabla en html viene marcada por las etiquetas **<table>** y **</table>**. Entre esas dos etiquetas definiremos las celdas, las columnas y las características de cada uno de estos parámetros.

En la etiqueta **<table>** podemos definir el fondo; el parámetro **"bgcolor"** corresponde al color de fondo y el parámetro **"background"** coloca una imagen de fondo.

Otro aspecto que podemos definir de la tabla es el borde. Esto lo haremos con el parámetro **"border"**. Si no ponemos borde, escribimos "0". Para darle color al borde, utilice la etiqueta **"bordercolor"** e indicando el color que queramos para nuestro borde.

El parámetro **"width"** indicará el ancho de la tabla. Este parámetro recibe valores en píxeles o con porcentaje.

Dos parámetros más son **cellspacing** (que define el espacio entre las celdas de la tabla) y **cellpadding** (que le marca a la tabla el espacio que debe dejar alrededor del texto dentro de una celda).

### Etiqueta fila: <tr>

El encabezado las filas se escriben a través de las etiquetas <tr></tr>. El contenido de las columnas que están dentro de la fila lo podemos alinear tanto horizontal como verticalmente.

Para alinear verticalmente utilizaremos el atributo "valign": alinear arriba de la celda ("top"), en el centro ("middle") o debajo ("bottom").

Para alinear horizontalmente utilizaremos el atributo "align": alinear el contenido de las celdas en el centro ("center"), a la izquierda ("left"), a la derecha ("right") o justificado ("justify").

### Etiqueta celdas <td>

Las celdas que van dentro de cada fila se definen con la etiqueta <td></td>.

Al igual que en las filas, en las celdas podemos definir la alineación del contenido que está dentro con los atributos "valign" y "align". Las celdas poseen atributos que nos ayudan a agrupar celdas o columnas. Para agrupar celdas utilizaríamos el atributo "colspan" y para agrupar filas el atributo "rowspan".

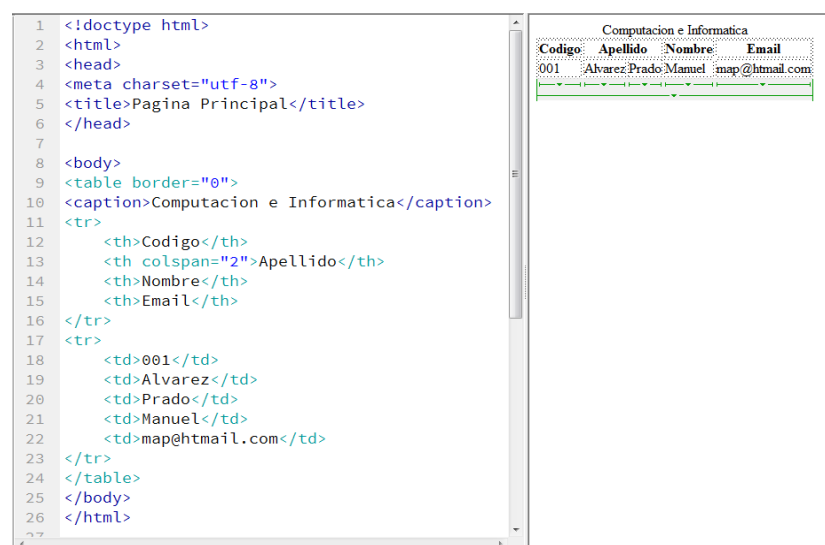
### Etiqueta de encabezado <th>

La etiqueta <th></th> está considerado como el encabezado de la tabla, por lo que se creará en negrita y centrado.

### Etiqueta "caption"

Permite colocar un título o encabezado a la tabla. Puedes colocar el encabezado arriba o abajo mediante la etiqueta "align": "align=top" para arriba y "align=bottom" abajo.

Acontinuación presentamos un ejemplo utilizando tablas.

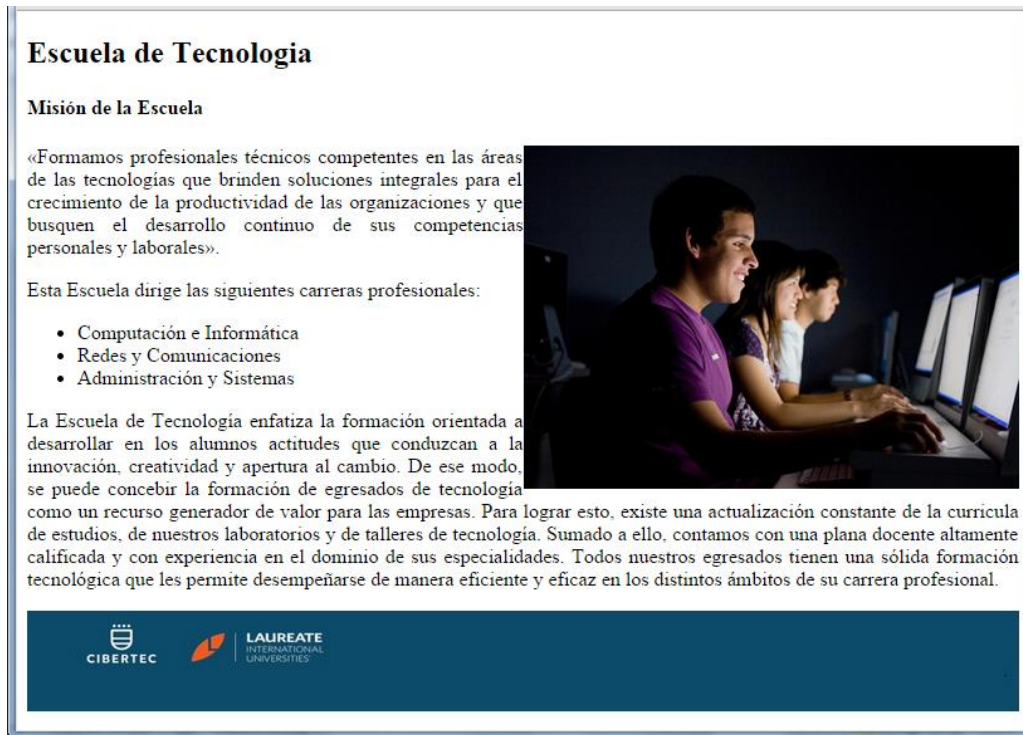




## LABORATORIO 3.1

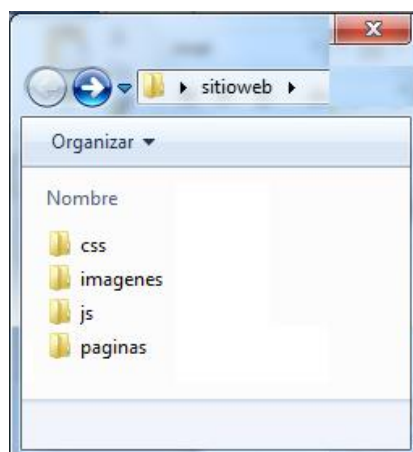
### Trabajando con Etiquetas

Se pide diseñar una página web, tal como se muestra, que permita describir la misión de la Escuela de Tecnología del Instituto Superior Cibertec.

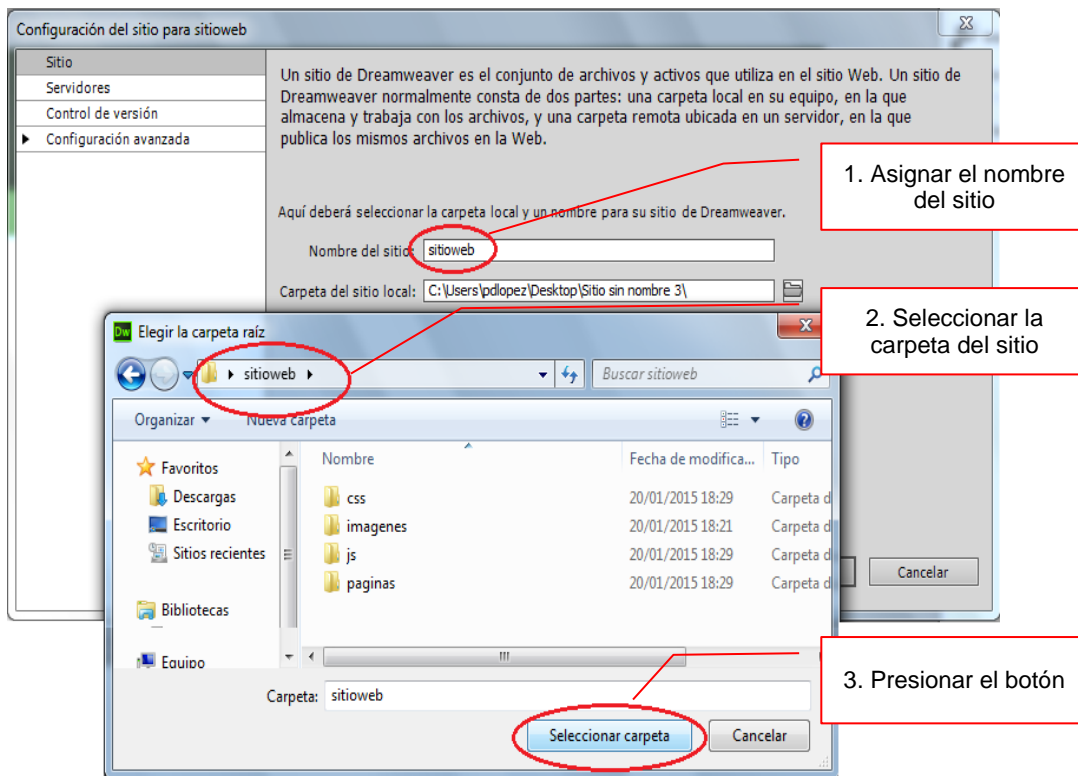


### 1. Creando el Sitio Web

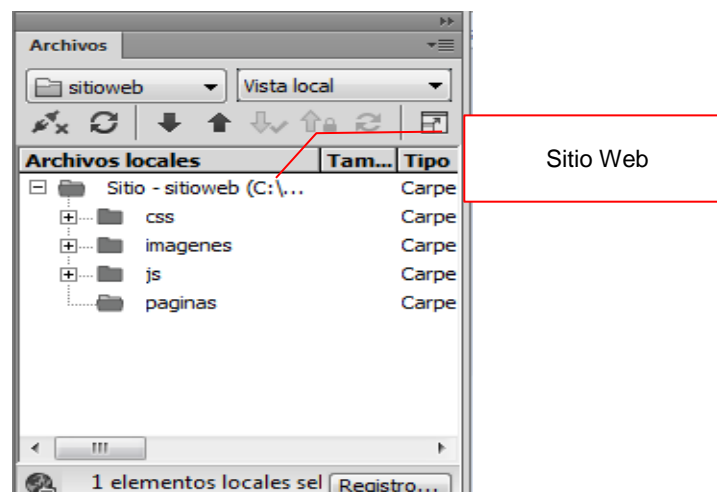
Defina una carpeta llamado sitioWeb, en ella crea las siguientes sub carpetas, tal como se muestra.



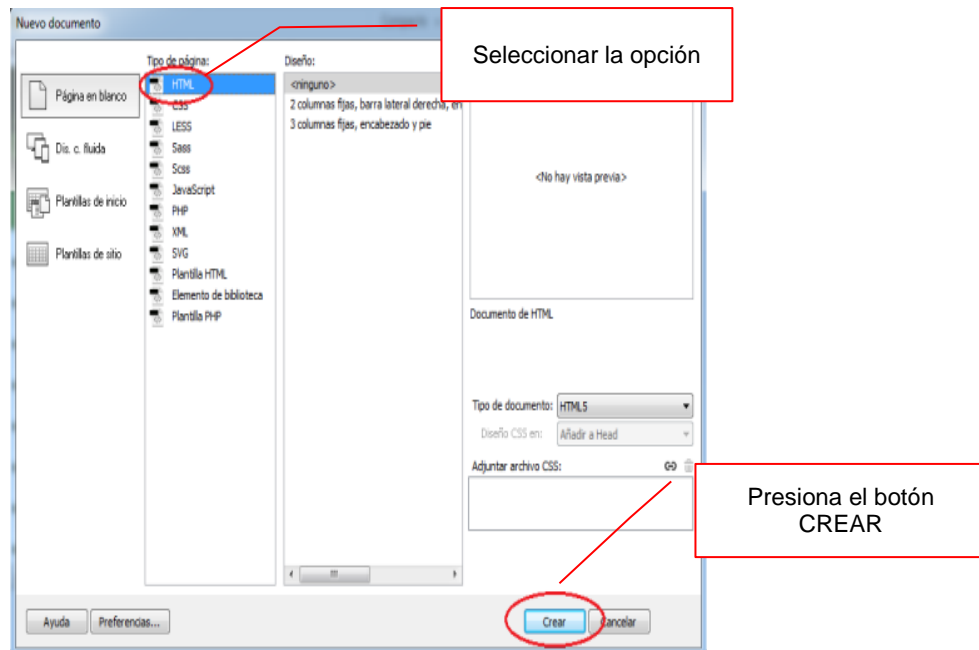
En el aplicativo DreamWeaver, configuramos un sitio web, tal como se muestra



Configurado el sitio web, este se visualice en la ventana Archivos, tal como se muestra en la figura.

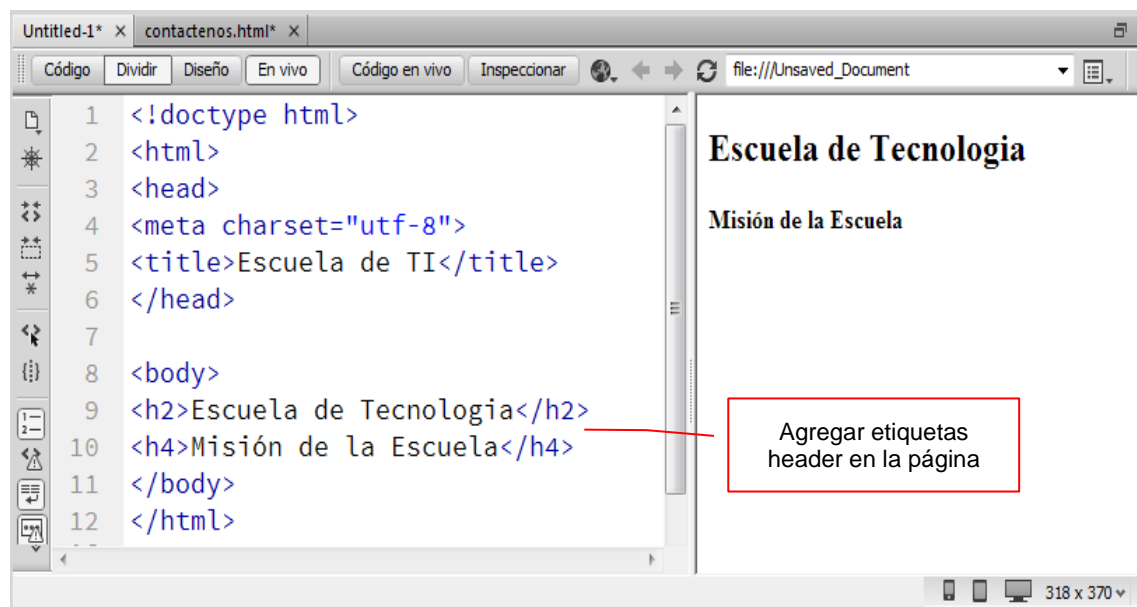


A continuación creamos un nuevo documento. Desde la opción **Archivo** selecciona Nuevo Documento. Selecciona el tipo de página: HTML, y presiona el botón CREAR.



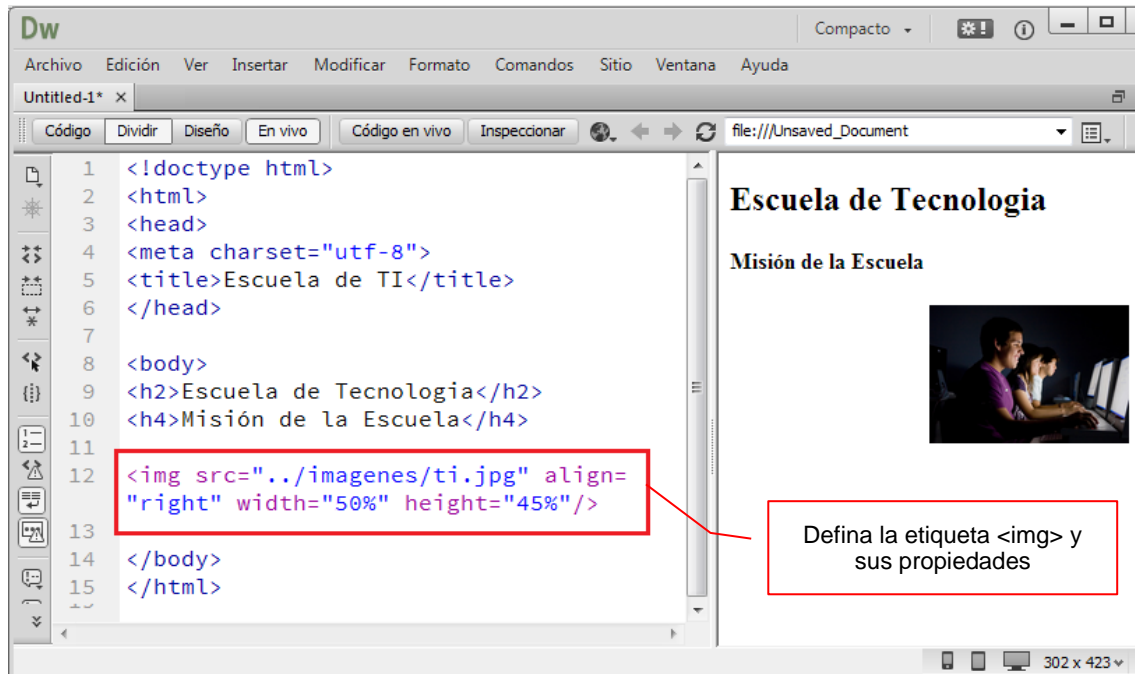
## Añadiendo etiquetas header

A continuación agregamos etiquetas head `<h2>` y `<h4>`, tal como se muestra. Presiona la tecla F5 para visualizar la vista de la página



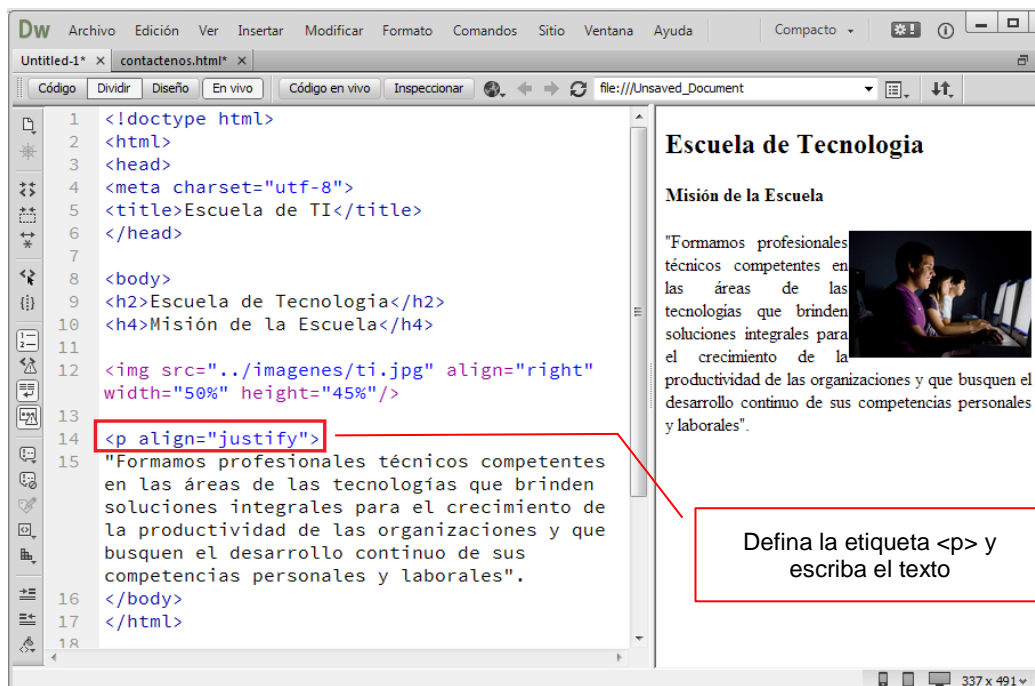
## Agregando la imagen de la Escuela

Para incrustar una imagen en la página, defina la etiqueta <img> indicando el origen de la imagen: src, tal como se muestra, definida la etiqueta presiona la tecla F5 para visualizarla.



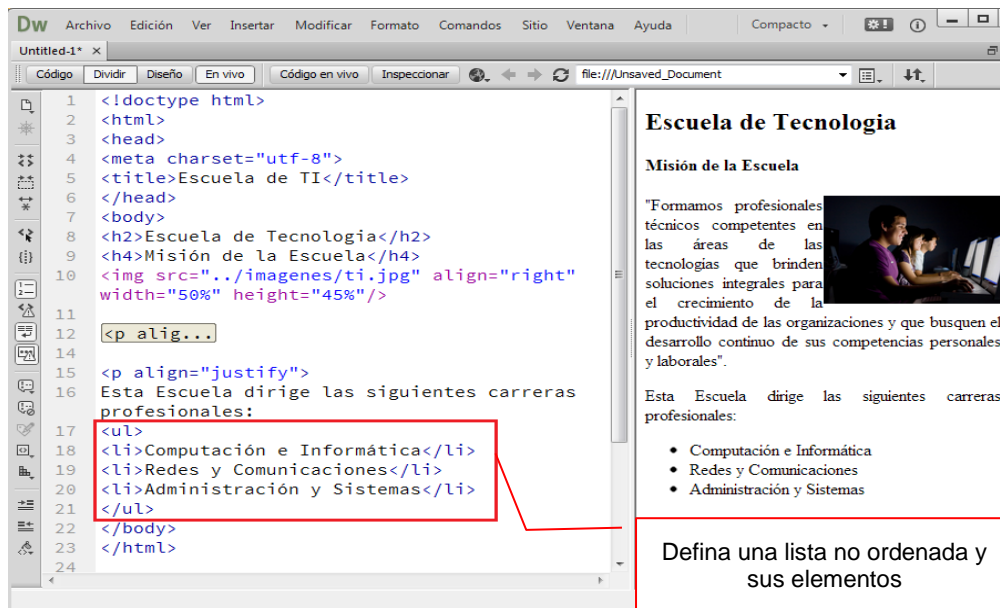
## Agregando texto a la página

Acontinuación defina la etiqueta <p> (párrafo de alineamiento justificado), digite el texto siguiente en la página, tal como se muestra.



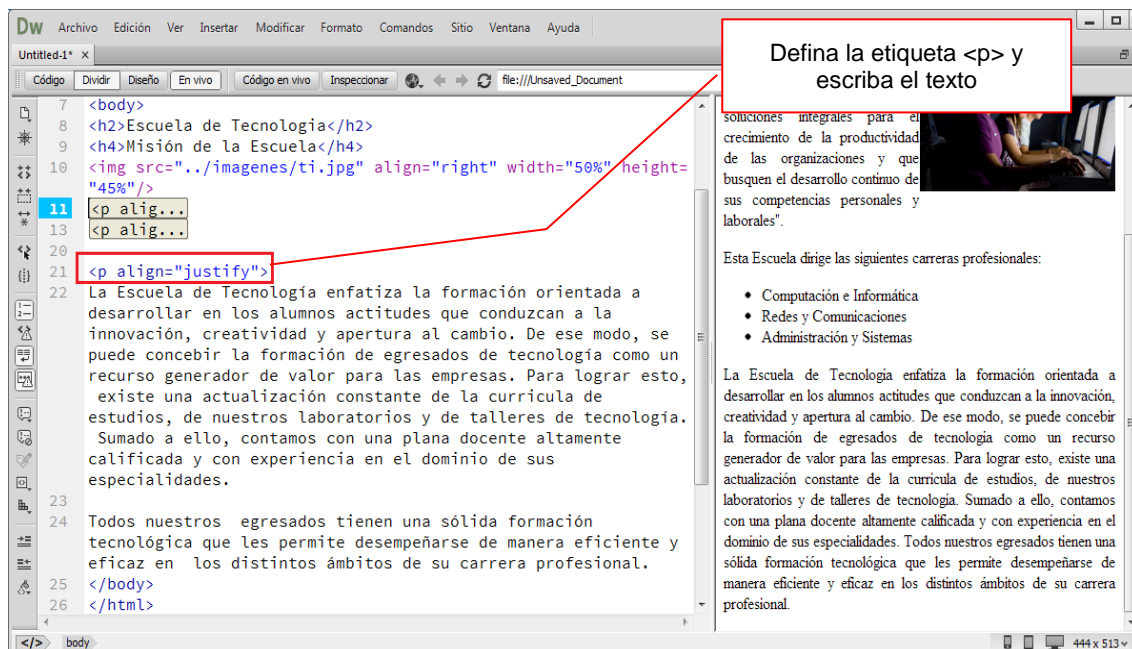
## Agregando listas

A continuación defina una lista no ordenada <ul> y sus elementos <li>, para mencionar las carreras de la Escuela, tal como se muestra para



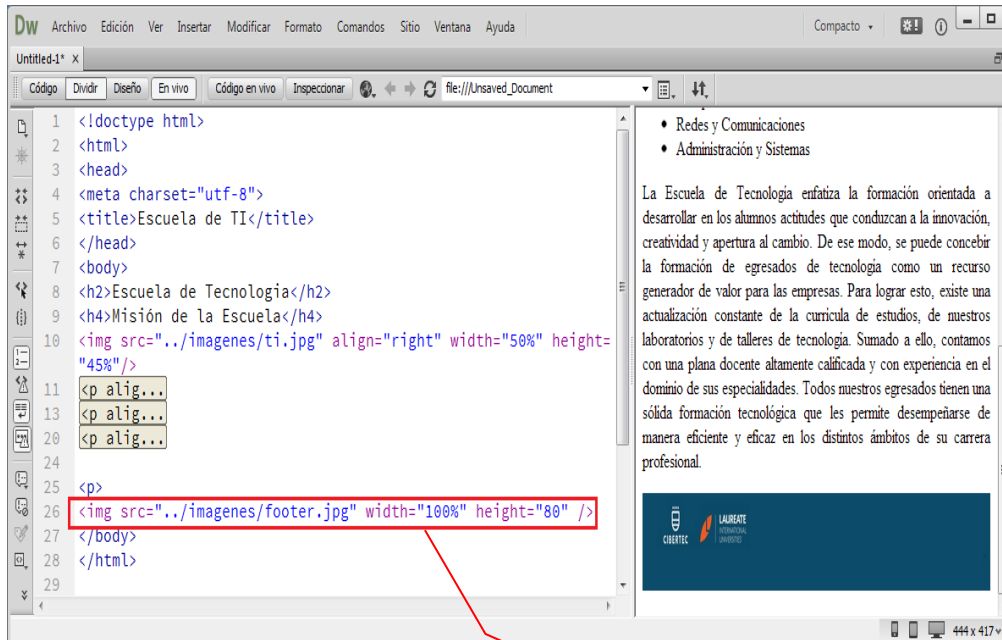
## Agregando texto a la página

A continuación defina la etiqueta <p> (párrafo de alineamiento justificado), digite el texto siguiente en la página, tal como se muestra



## Agregando la imagen del pie de página

Para incrustar una imagen en la página, defina la etiqueta <img> indicando el origen de la imagen: src, tal como se muestra, definida la etiqueta presiona la tecla F5 para visualizarla.



Defina la etiqueta <img>  
para el pie de página

Para visualizar la página presiona la tecla F12, ejecutando la página en un navegador seleccionado.





## LABORATORIO 3.2

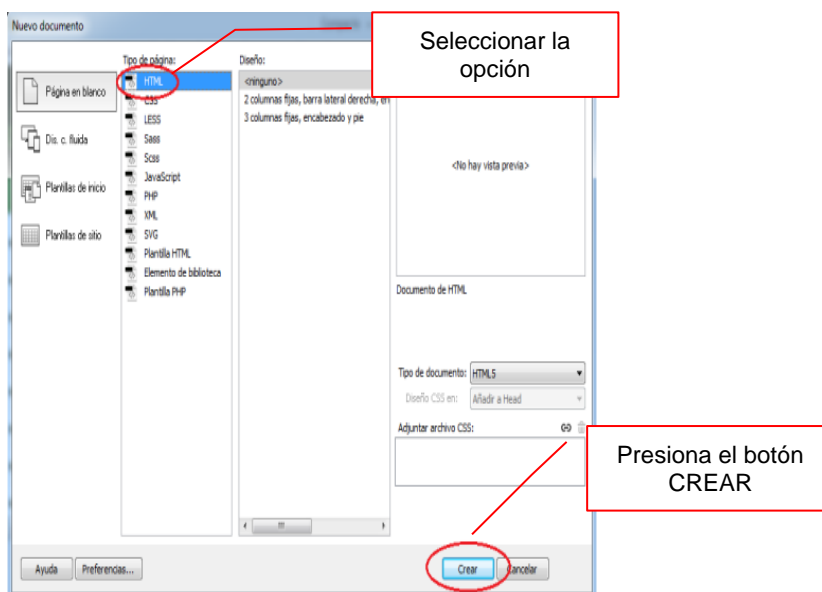
### Trabajando con Etiquetas

Se pide diseñar una página web que permita describir la carrera de Administracion y Sistemas de la Escuela de Tecnologia del Instituto Superior Cibertec, tal como se muestra en la figura.

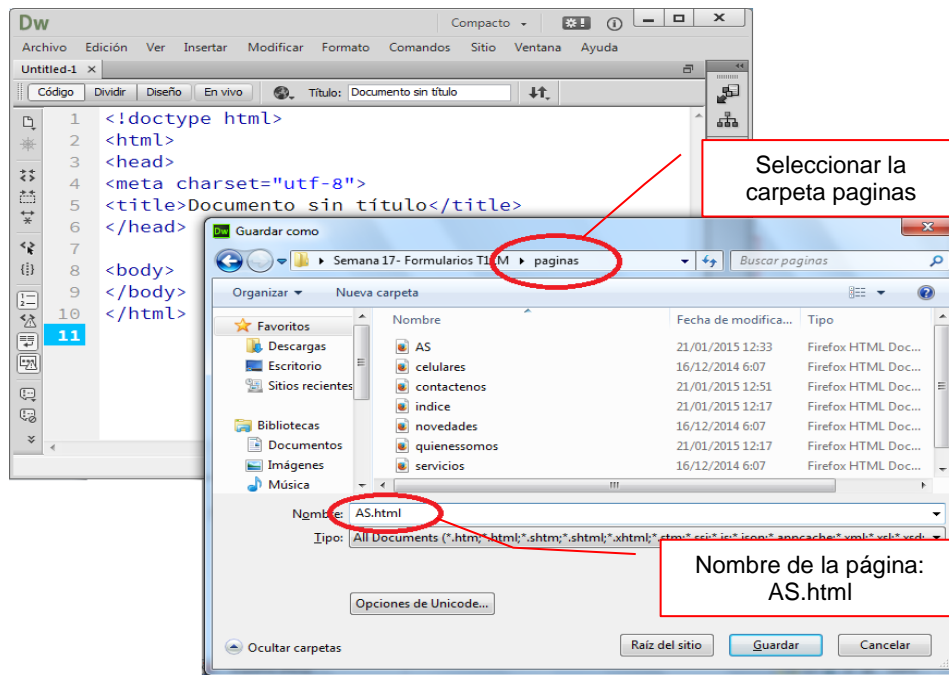


### Agregar una página HTML y guardarla

Acontinuación creamos un nuevo documento. Desde la opción **Archivo** selecciona Nuevo Documento. Selecciona el tipo de página: HTML, y presiona el botón CREAR.

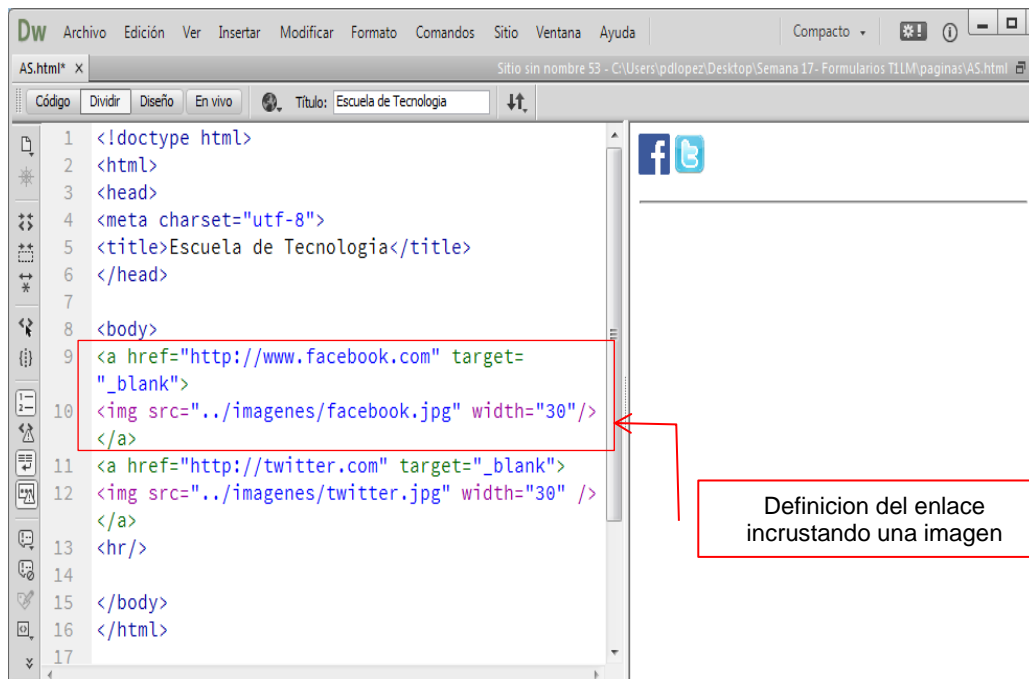


Para guardar el documento, presiona la combinación de las teclas CTRL + S: selecciona la carpeta paginas, y asigne el nombre del archivo: AS.html. presiona la tecla Guardar.



### Agregando enlaces

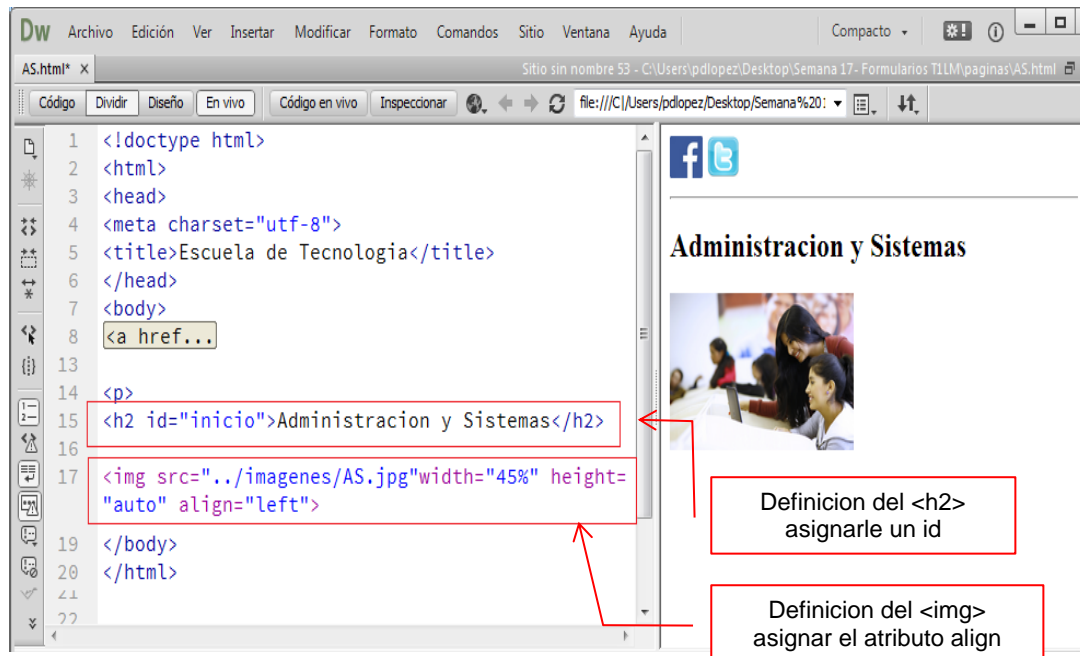
Para mostrar los enlaces a las redes sociales a través de imágenes, primero defina la referencia a la página <a href> y dentro de la etiqueta <a>, agrega la etiqueta <img>. Al colocar los enlaces a las redes sociales, coloque una línea: <hr>





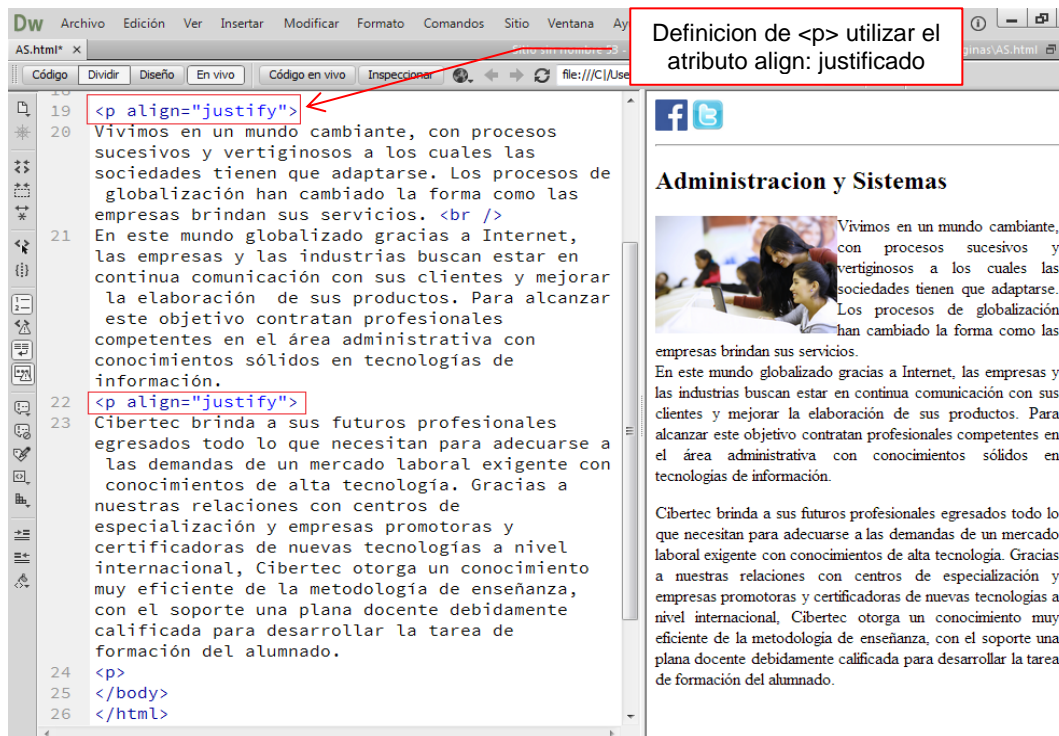
## Agregando header e imagen

A continuación, defina una etiqueta headering `<h2>` cuyo `id="inicio"`, lo utilizaremos para redireccionar la pagina de manera interna; luego agregar la etiqueta `<img>` la cual representa a la carrera de Administracion y Sistemas.



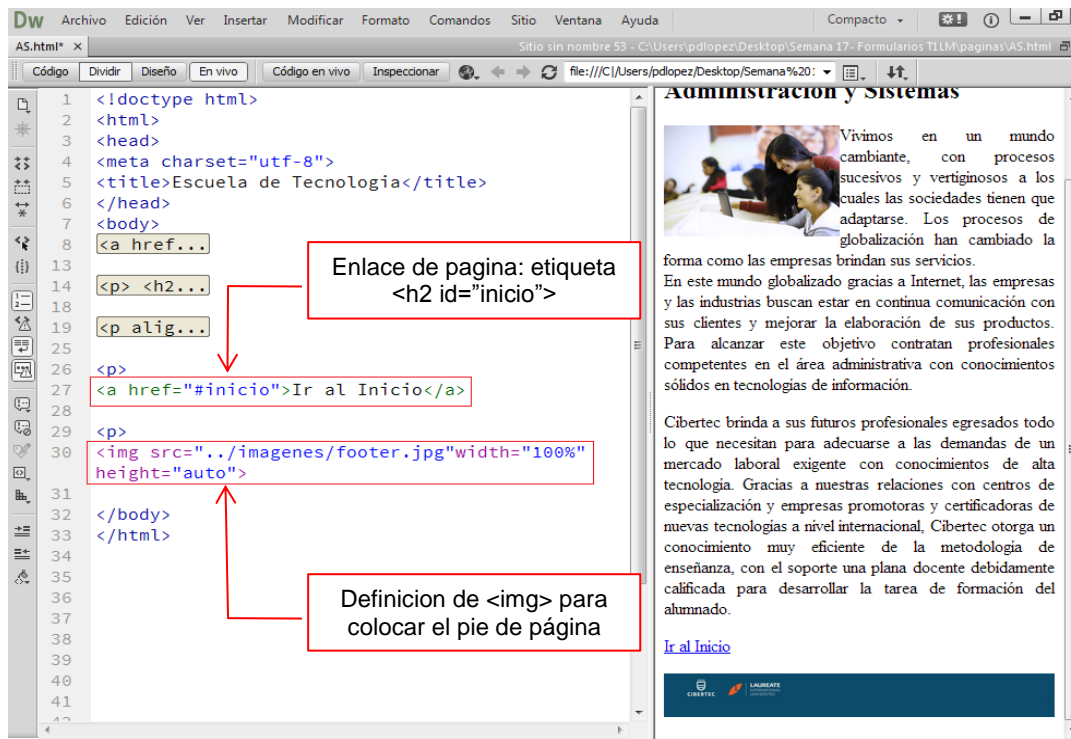
## Agregando texto a la página

A continuación defina la etiqueta `<p>` (párrafo de alineamiento justificado), digite el texto siguiente en la página, tal como se muestra



## Agregando referencia de página e imagen del pie de página

Acontinuación agregamos una etiqueta <a> para hacer una referencia de pagina hacia la etiqueta <h2> (**ver al inicio del ejercicio**). Para incrustar una imagen en la página, defina la etiqueta <img> indicando el origen de la imagen: src, tal como se muestra, definida la etiqueta presiona la tecla F5 para visualizarla




Para visualizar la página presiona la tecla F12, ejecutando la página en un navegador seleccionado




# Resumen


- 📖 Ante las necesidades de los usuarios y productores de contenido se ha lanzado la versión HTML5. Esta nueva versión del lenguaje Web simplifica el trabajo y agrupa las nuevas tecnologías de desarrollo de aplicaciones web: CSS3 y Javascript.
- 📖 **Dreamweaver** es el mejor editor para crear páginas web, permite crear avanzadas web que se pueden visualizar también en dispositivos móviles, sin tener que dominar el código html, combina las últimas tecnologías en desarrollo web como Javascript, CSS, AJAX, XHTML, Adobe AIR, Smart Objects de Photoshop, subversiones (SVN), frameworks Javascript y más.
- 📖 Un sitio Web ofrecerá tanto a clientes como a proveedores y empleados, un canal de comunicación abierto las 24 hrs. del día con información completa y actualizada sobre la organización, para que realicen consultas, pedidos o sugerencias.
- 📖 Un documento HTML no es más que un archivo de texto. La extensión de su nombre suele ser .html o .htm. Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto, imágenes, videos, audio, etc.
- 📖 La estructura de un documento en HTML5 inicia con la declaración del tipo de documento que vamos a utilizar, esto lo hacemos definiendo la etiqueta **doctype**. Para construir la estructura del documento HTML definimos las etiquetas **<html></html>**. La palabra head viene del inglés “cabeza” y su función es delimitar cabecera del documento. Sus etiquetas son: **<head>** y **</head>**. La etiqueta **<body>** delimita el cuerpo del documento HTML. Este elemento contiene todo el contenido visible de un documento HTML (párrafos de texto, imágenes, tablas).
- 📖 HTML agrega automáticamente un espacio antes y después de cada título. **<h1>** al ser usado como título de una página web, es de suma importancia ya que es uno de los parámetros de Google y demás buscadores tiene en cuenta a la hora de indexar un sitio web.
- 📖 La etiqueta párrafo se define con la etiqueta **<p>**. Esta etiqueta permite definir los párrafos que forma el texto de una página. El texto del párrafo lo puedes alinear utilizando la etiqueta **align** utilizando los parámetros “center”, “right”, “left” y “justify”.
- 📖 La etiqueta **<br>** incluye una nueva línea y fuerza a que el texto se muestre en la siguiente línea. Es el equivalente a presionar la tecla ENTER (o Intro).
- 📖 Los enlaces en HTML se crean mediante la etiqueta **<a>** (viene del inglés “anchor” traducido literalmente sería “ancla”). El atributo más importante de la etiqueta **<a>** es **href**, que se utiliza para indicar que apunta a ese enlace. Las URL de los enlaces pueden ser: absolutas, relativas, internas y externas.
- 📖 Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta **<img>** y las imágenes de adorno no se deberían incluir en el código HTML, sino que deberían emplearse hojas de estilos CSS para mostrarlas. Los dos atributos requeridos son src y alt. El atributo src representa la dirección URL de la imagen; la URL pueden ser absoluta o relativa. El atributo alt permite describir el contenido de la imagen mediante un texto breve.
- 📖 Las etiquetas **<figure>** y **<figcaption>** permiten definir una mejor semántica (significado o sentido) a nuestro contenido dentro de un documento HTML5.
- 📖 Las listas nos permiten ordenar o enumerar datos o elementos para facilitar su lectura. Tenemos tres tipos de listas:
  - No ordenadas.

- Ordenadas.
- De definición.

 Las tablas son herramientas muy útiles para presentar datos en una estructura de filas y columnas, para el diseño de páginas y ubicación de textos y gráficos dentro de las mismas. Una tabla en html viene marcada por las etiquetas <table></table>. Entre esas dos etiquetas definiremos las las celdas, las columnas y las características de cada uno de estos parámetros.

 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

 <http://www.virtualnauta.com/html-formato-de-texto>

 [http://librosweb.es/xhtml/capitulo\\_3/marcado\\_avanzado\\_de\\_texto.html](http://librosweb.es/xhtml/capitulo_3/marcado_avanzado_de_texto.html)

 <http://html.hazunaweb.com/103.php>

 [https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5\\_lista\\_elementos](https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos)



# HTML5 Y CSS3

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3.

## TEMARIO

### 2.1 Tema 3: Estructura HTML y aplicación de estilos CSS

2.1.1 Entorno del programa (Editor HTML)

2.1.2 Creación de un sitio web (estructura de carpetas)

2.1.3 Estructura de un documento HTML

2.1.3.1 Estructura del documento

2.1.3.2 Etiquetas básicas: encabezados (H1...H6), párrafos, listas, enlaces internos y externos, imagen.

### 2.2 Tema 4: Estilos CSS

#### 2.2.1 Estilos CSS

2.2.1.1 Tipos de Estilos: En línea e internos.

2.2.1.2 Estilos aplicados a etiquetas y de clase.

## ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5 y hojas de estilo CSS3.
- Los alumnos diseñan páginas web utilizando bloques de contenidos.



## 1.7 ESTILOS CSS

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, las cuales están escritas con código HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.



Al crear una página web, se utiliza en primer lugar el lenguaje HTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

### Características y ventajas de las CSS

Para definir las CSS éstas se hacen mediante una sintaxis especial, la cual se aplica a:

- Un sitio web entero, de modo que se puede definir la forma de todo el web de una sola vez.
- Un documento HTML o página, se puede definir la forma, en una pequeña porción de código en la cabecera, a toda la página.
- Una porción del documento, aplicando estilos visibles en una porción de la página.
- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación.

La potencia de la tecnología salta a la vista. Si antes el HTML se nos quedaba corto para maquetar las páginas y teníamos que utilizar trucos para conseguir nuestros efectos, ahora tenemos muchas más herramientas que nos permiten definir esta forma:

- Podemos definir la distancia entre líneas del documento.
- Se puede aplicar identado a las primeras líneas del párrafo.
- Podemos colocar elementos en la página con mayor precisión, y sin lugar a errores.

- Y mucho más, como definir la visibilidad de los elementos, márgenes, subrayados, tachados.

Además, utilizando CSS podemos definir las dimensiones de los elementos de la página utilizando muchas más unidades como:

- Pixels (px) y porcentaje (%), como antes.
- Pulgadas (in)
- Puntos (pt)
- Centímetros (cm)

### Soporte de CSS en los Navegadores

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador. En la siguiente tabla se muestra el soporte de CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los navegadores.

Navegador	Motor	CSS 2.1	CSS 3
Google Chrome	WebKit	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Internet Explorer	Trident	Completo	Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador
FireFox	Gecko	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Safari	WebKit	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Todos los selectores, pseudo-clases y muchas propiedades

Referencia: [http://librosweb.es/css/capitulo\\_1/soporte\\_de\\_css\\_en\\_los\\_navegadores.html](http://librosweb.es/css/capitulo_1/soporte_de_css_en_los_navegadores.html)

### Funcionamiento de las CSS

CSS funciona en base a declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por un selector, una propiedad y el valor que se le asigne.

```
h1{
    color: black;
    font-style: italic
}
```



Donde:

h1 es el selector

{color: black;} es la declaración, **color** es la propiedad y **black** es el valor

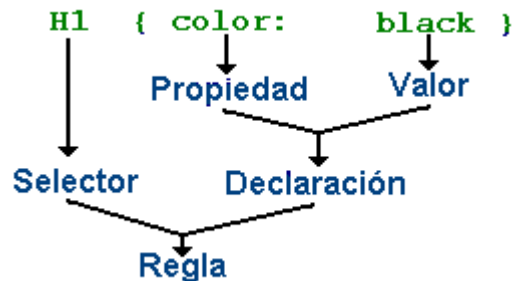


Figura 1

Referencia: <http://tejedoresdelweb.com/w/CSS>

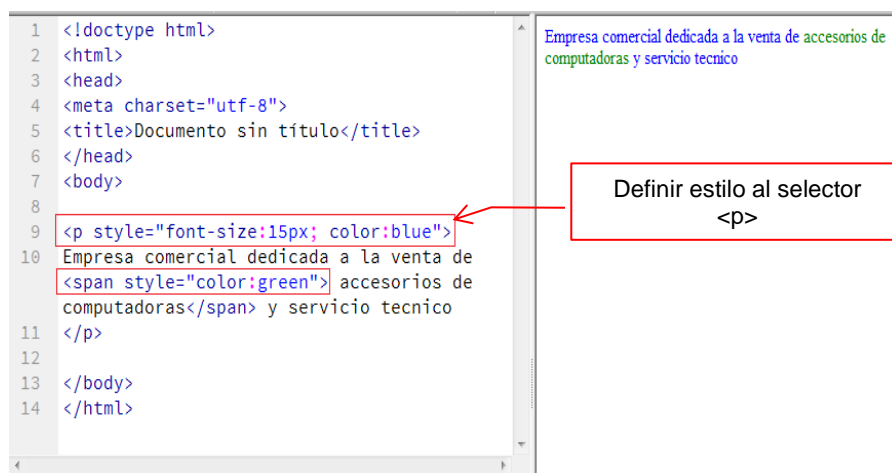
El **selector** funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La **declaración** es la parte de la regla que establece cuál será el efecto.

En el ejemplo anterior, el selector **h1** indica que todos los elementos de tipo **h1** se verán afectados por la declaración donde se establece que la propiedad color va a tener el valor red (rojo) para todos los elementos **h1** del documento o documentos que estén vinculados a esa hoja de estilos.

### 1.9.1 Tipos de Estilos

#### Estilo en línea

Es el estilo menos utilizado por que incluye la CSS en los mismos elementos de documento HTML.

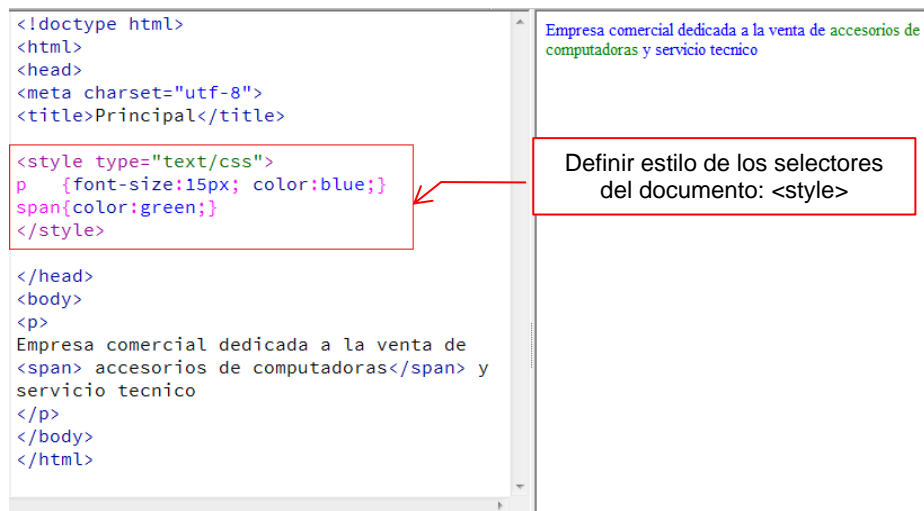


Este estilo no es muy utilizado por cuanto la regla solamente se aplica a un selector específico dentro del mismo documento.

#### Estilo interno

Si deseamos crear definiciones de estilo solamente para una página del sitio web debemos crear un estilo interno.

Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta **<style>** de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección **<head>**).



Este método se emplea cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos que se incluyen por defecto en todas las páginas del sitio web.

El principal inconveniente es que si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

### Estilo externo

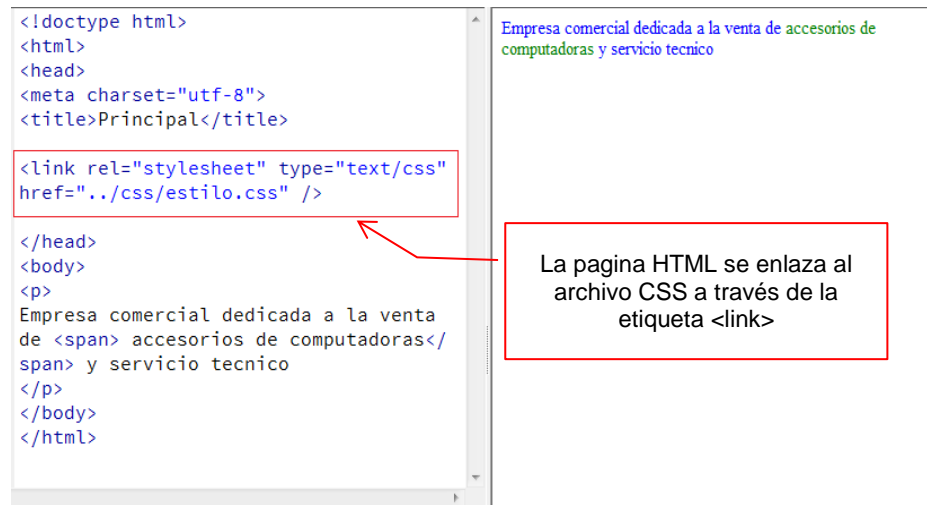
Si deseamos que el sitio Web tenga un aspecto uniforme debemos utilizar una hoja de estilos externa y vinculada a cada documento. Las hojas de estilo se pueden almacenar en **archivos externos** y vincularlas a uno o más documentos.



Una hoja de estilo externa es un archivo que contiene únicamente especificaciones CSS. Podemos utilizar hojas de estilo externas para asegurar una homogeneidad de diseño entre las páginas.

Un archivo de tipo CSS se podrá enlazar a una página HTML mediante la etiqueta `<link>`. Un archivo de tipo CSS no es más que un archivo simple

de texto cuya extensión es .css Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.



La etiqueta **<link>** incluye los siguientes atributos cuando enlaza un archivo CSS:

- **rel**: indica el tipo de relación que existe entre el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor **stylesheet**
- **type**: indica el tipo de recurso enlazado. Sus valores están estandarizados y para los archivos CSS su valor siempre es **text/css**
- **href**: indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.

Para incluir en la página HTML los estilos definidos en archivos CSS externos se utiliza una regla especial de tipo **@import**. Las reglas de tipo **@import** siempre preceden a cualquier otra regla CSS (con la única excepción de la regla **@charset**).

La URL del archivo CSS externo se indica mediante una cadena de texto encerrada con comillas simples o dobles o mediante la palabra reservada **url()**. De esta forma, las siguientes reglas **@import** son equivalentes:

```
@import '/css/estilos.css';
@import "/css/estilos.css";
@import url('/css/estilos.css');
@import url("/css/estilos.css");
```

### 1.9.2 Estilos aplicados a etiquetas y de clase

#### Selector universal

Se utiliza para seleccionar todos los elementos de la página. El selector universal se indica mediante un asterisco (\*).

A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

El siguiente ejemplo elimina el margen, el borde y el relleno de todos los elementos HTML.

```
*{  
    margin:0;  
    border:0px;  
    padding:0;  
}
```

#### Selector de tipo o etiqueta

Este tipo de selector identifica a un tipo de elemento que conforma el código HTML modificando el aspecto predeterminado de la etiqueta.

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p{  
    color:red;  
    text-align:justify;  
    font-style:italic;  
}
```

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores. En el siguiente ejemplo, los títulos de sección h1, h2 y h3 comparten los mismos estilos:

```
h1, h2, h3{  
    color:blue;  
    text-align:left;  
    text-decoration:underline;  
    font-style:italic;  
}
```

En las hojas de estilo complejas, es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos.

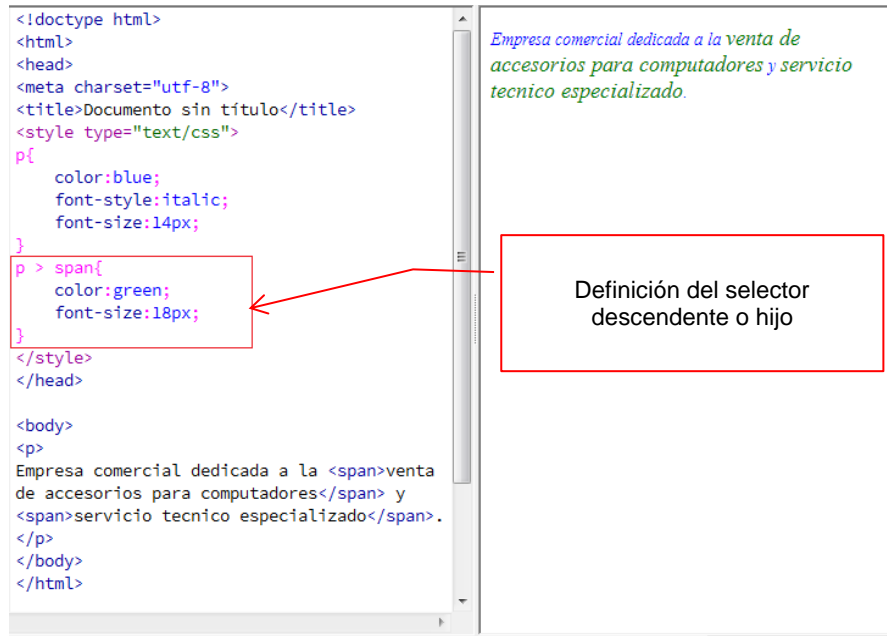
#### Selector Descendente

Este tipo de selector, selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

```
p > span{  
    color:green;  
    font-size:18px;  
}
```

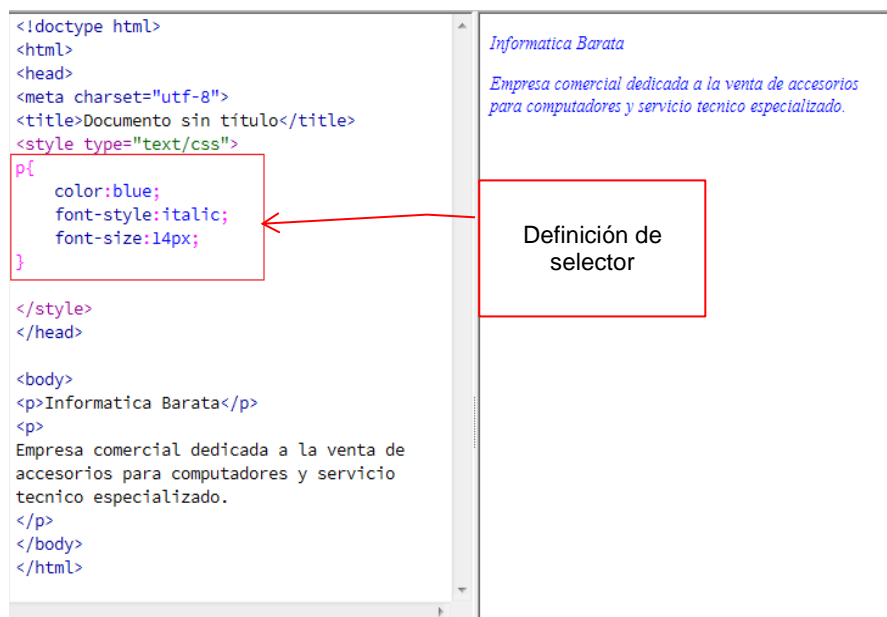
Se utiliza para seleccionar un elemento que es hijo de otro elemento y se indica mediante el "signo de mayor que" (>).

Mientras que en el selector descendente sólo importa que un elemento esté dentro de otro, independientemente de lo profundo que se encuentre, en el selector de hijos el elemento debe ser hijo directo de otro elemento.



### Selector de Clase

Cuando se declara una regla en la hoja de estilo, esta regla se aplica a todos los elementos HTML descritos por su selector; por ejemplo:



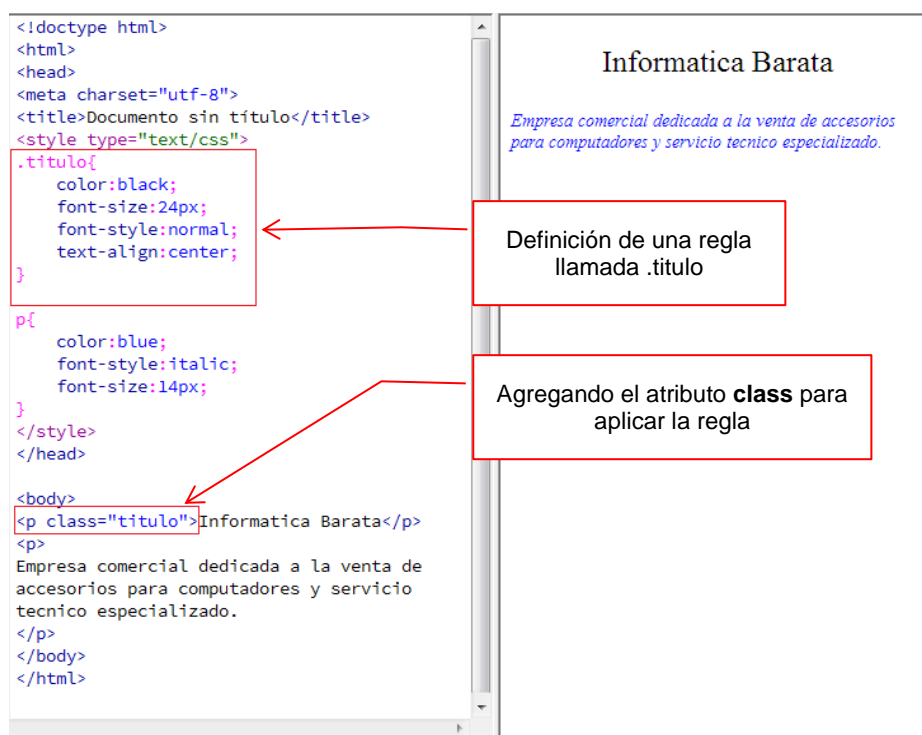
¿Qué sucede si en ciertos casos queremos usar un párrafo con características diferenciadas?

Este planteamiento se resuelve en utilizar el atributo **class** de HTML, para definir un nuevo selector. El selector de clases consta de un punto (.) seguido por el nombre de la clase que hayamos creado.

A continuación, defina una nueva regla llamada **titulo** con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo **class** con un punto (.) tal y como muestra el siguiente

```
.titulo{
    color:black;
    font-size:24px;
    font-style:normal;
    text-align:center;
}
```

Para aplicar el estilo de la clase CSS sobre un elemento utilizamos el atributo **class** junto con el nombre de la regla:



### Selector de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página HTML. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

```
#titulo{
    color:black;
    font-size:24px;
    font-style:normal;
    text-align:center;
}
```

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
<style type="text/css">
#titulo{
    color:black;
    font-size:24px;
    font-style:normal;
    text-align:center;
}

p{
    color:blue;
    font-style:italic;
    font-size:14px;
}
</style>
</head>
<body>
<p id="titulo">Informatica Barata</p>
<p>
    Empresa comercial dedicada a la venta de
    accesorios para computadores y servicio
    tecnico especializado.</p>
<p>
    Nuestra vision es ser lider en la venta de
    accesorios de computadoras a nivel Lima y
    provincias.</p>
</body>
</html>
```

Definición de una regla llamada #titulo

Agregando el atributo ID para aplicar la regla

Informatica Barata

*Empresa comercial dedicada a la venta de accesorios para computadores y servicio tecnico especializado.*

*Nuestra vision es ser lider en la venta de accesorios de computadoras a nivel Lima y provincias.*

### Pseudo-clases

Las pseudo-clases permiten al diseñador web variar los estilos de un elemento en respuesta a las acciones del usuario. A continuación se indican las acciones del usuario que activan cada pseudo-clase:

- **:link**, especifica el aspecto de los enlaces que todavía no han sido visitados.
- **:visited**, especifica el aspecto de los enlaces que sí han sido visitados.

- **:hover**, se activa cuando el usuario pasa el mouse o cualquier otro elemento apuntador por encima de un elemento.
- **:active**, se activa cuando el usuario activa un elemento, por ejemplo cuando pulsa con el ratón sobre un elemento. El estilo se aplica durante un espacio de tiempo prácticamente imperceptible, ya que sólo dura desde que el usuario pulsa el botón del ratón hasta que lo suelta.
- **:focus**, se activa cuando el elemento tiene el foco del navegador, es decir, cuando el elemento está seleccionado. Normalmente se aplica a los elementos `<input>` de los formularios cuando están activados y por tanto, se puede escribir directamente en esos campos.

Debido a esta característica y al comportamiento en cascada de los estilos CSS, es importante cuidar el orden en el que se establecen las diferentes pseudo-clases. El siguiente ejemplo muestra el único orden correcto para establecer las cuatro pseudo-clases principales en un enlace:

- `a:link { ... }`
- `a:visited { ... }`
- `a:hover { ... }`
- `a:active { ... }`



## LABORATORIO 4.1

### Trabajando con Estilos CSS

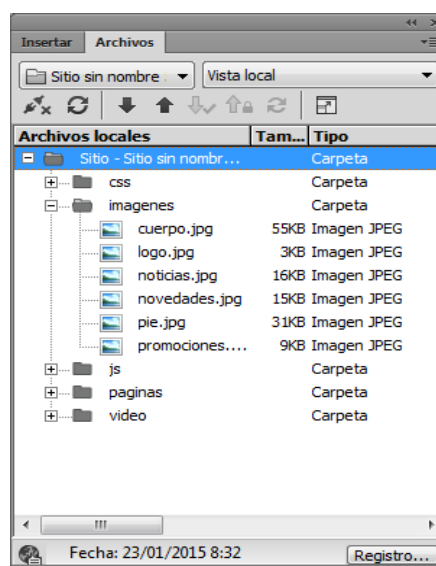
Se nos pide implementar una página para promocionar los productos y servicios de la empresa Toshiba, tal como se visualiza.

The screenshot shows the Toshiba website layout with the following annotated components:

- Direccionar a las redes sociales:** Points to the Facebook and Twitter social media icons in the top right corner.
- Logo (imagen):** Points to the large 'TOSHIBA' logo and the tagline 'Leading Innovation'.
- Enlaces:** Points to the navigation links: 'Productos Digitales', 'Dispositivos/Infraestructura', and 'Servicio Técnico Sobre Toshiba'.
- Imágenes que promocionar productos y servicios:** Points to the six product category tiles: 'TV INTELIGENTE, ELECCIÓN INTELIGENTE', 'PORTÁTILES Y TABLETS', 'DISCOS DUROS', 'EQUIPOS MULTIFUNCIÓN', 'TOSHIBA LIGHTING', and 'ACCESORIOS'.
- Pie de página (imagen):** Points to the footer area containing 'Accesibilidad', 'Declaración de privacidad', 'Términos y condiciones del sitio web', 'Mapa del sitio web', and 'Copyright © 2015 Todos los derechos reservados.'

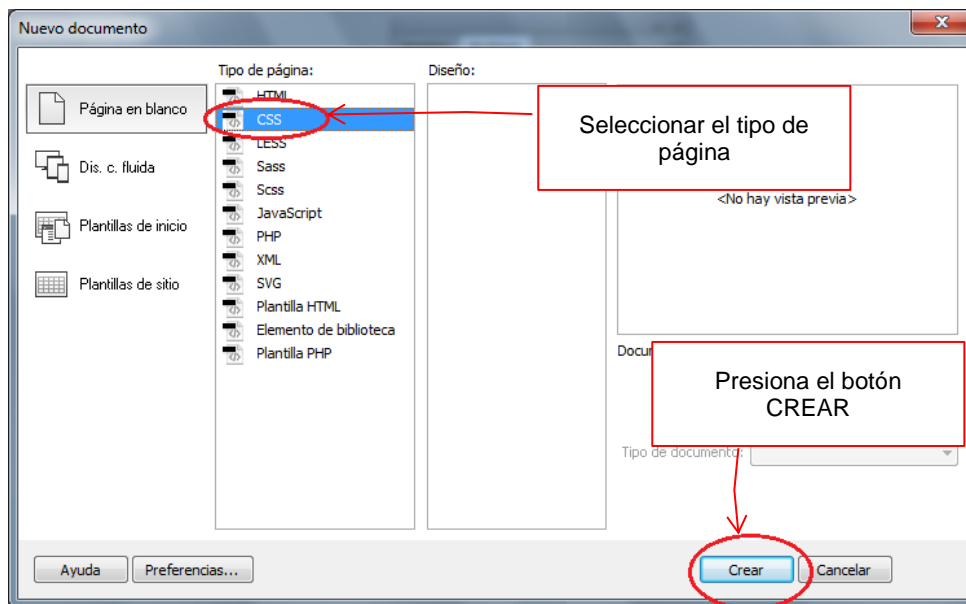
### Estructura del sitio Web

Defina la estructura de carpetas del sitio web, tal como se muestra. En la carpeta imágenes, hemos almacenado algunos archivos jpg para el diseño de la pagina HTML.

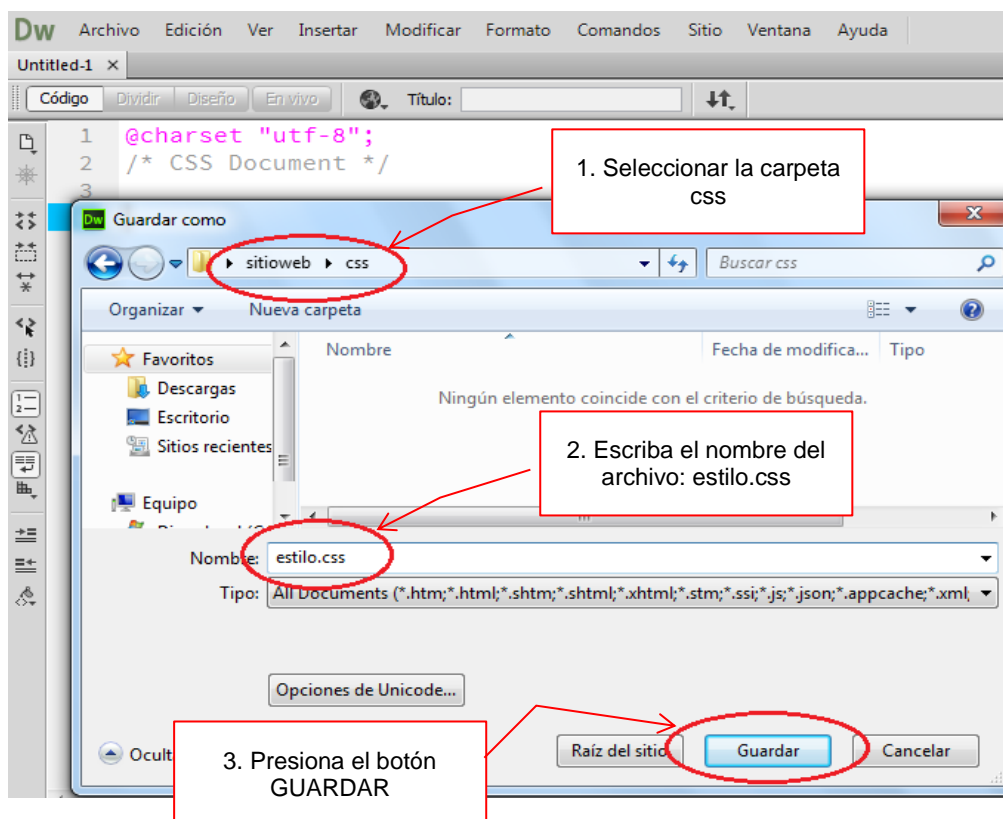


## Agregando un archivo CSS

Antes de iniciar con el diseño de la página HTML, primero creamos un archivo CSS, para definir las reglas de estilos que utilizará la página.

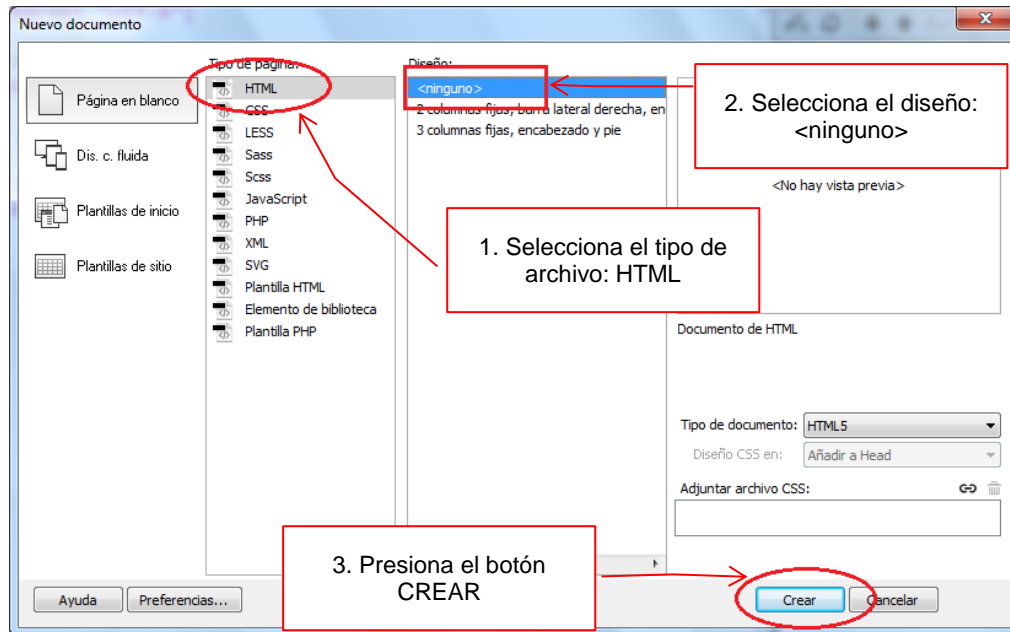


Creado el archivo, procedemos a guardarlo: presiona la combinación CTRL+S

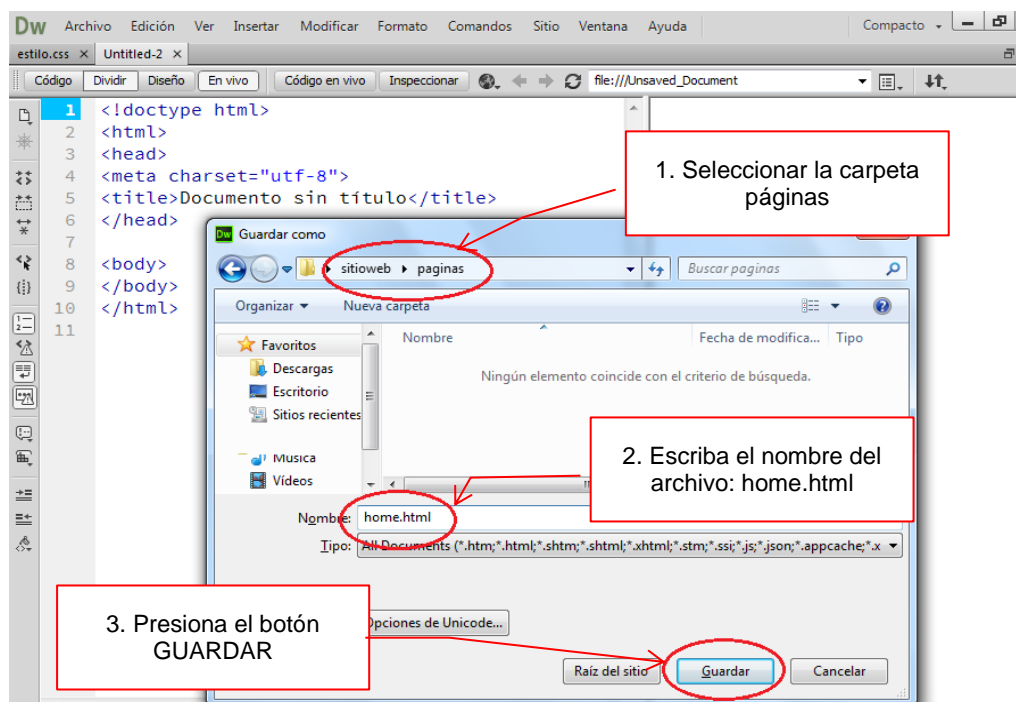


## Agregando la pagina HTML

A continuación, creamos un archivo HTML, el cual no tendrá diseño (ninguno), tal como se muestra

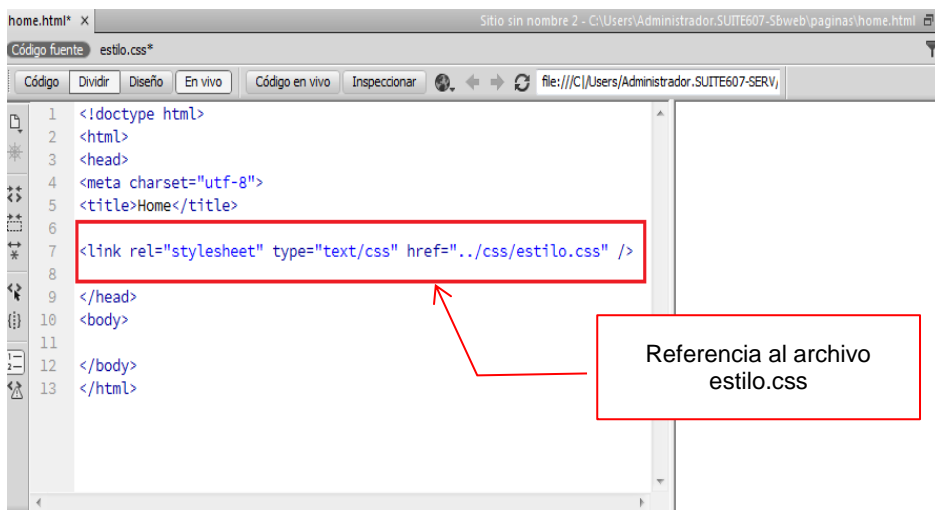


Creada el documento HTML, procedemos a guardarla, presiona la combinación CTRL+S



## Enlazando a la hoja de estilo

Creada la página, definimos, dentro del <head> la etiqueta <link> el cual hace una referencia al archivo CSS estilo.css, tal como se muestra.

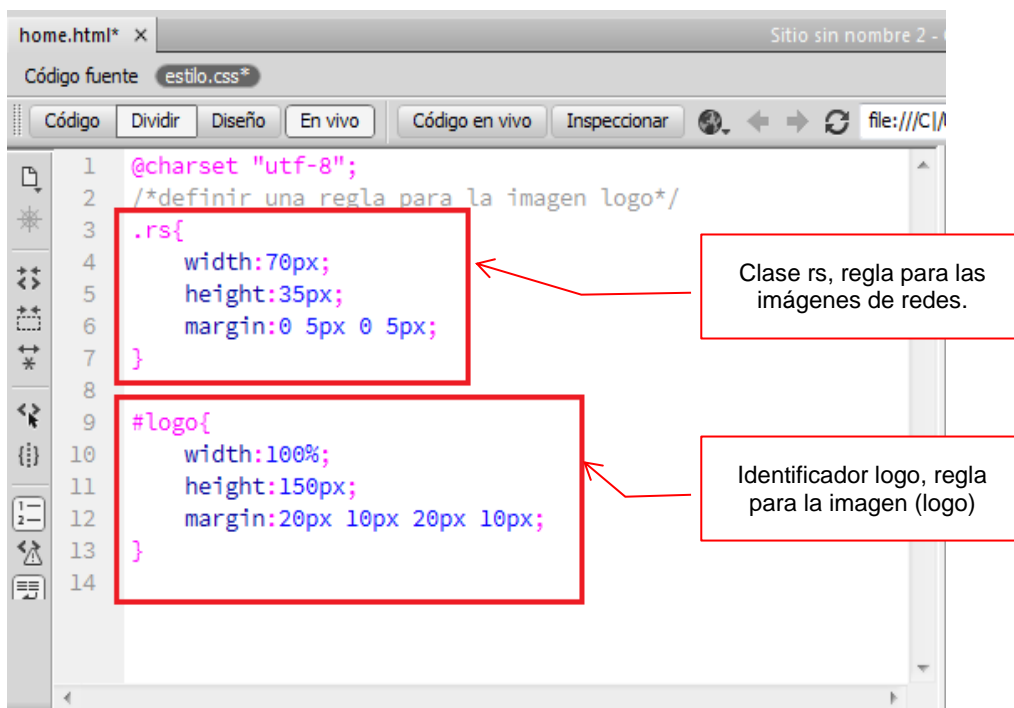


## Definiendo reglas de la sección logo y redes en el archivo estilo.css

En el archivo CSS, estilo.css definimos las primeras dos reglas: .rs y #logo

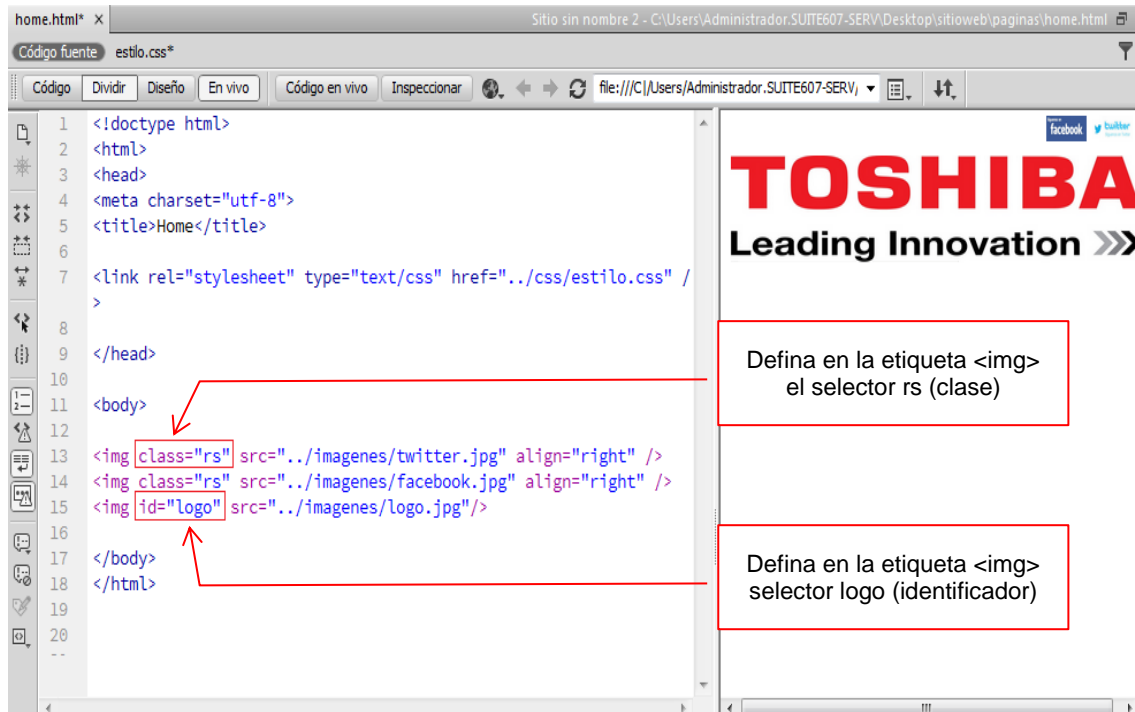
- Defina la clase **rs**, es una regla para las imágenes de redes sociales.
- El identificador **logo**, es una regla para la imagen de logo de la pagina.

Acontinuación guardar el archivo estilo.css: CTRL+S



## Agregando las etiquetas y reglas en la página

Definida las reglas en estilo.css, dentro del body defina las etiquetas <img> agregando a cada una su selector: de clase para las redes sociales y de identificador para el logo de la pagina.

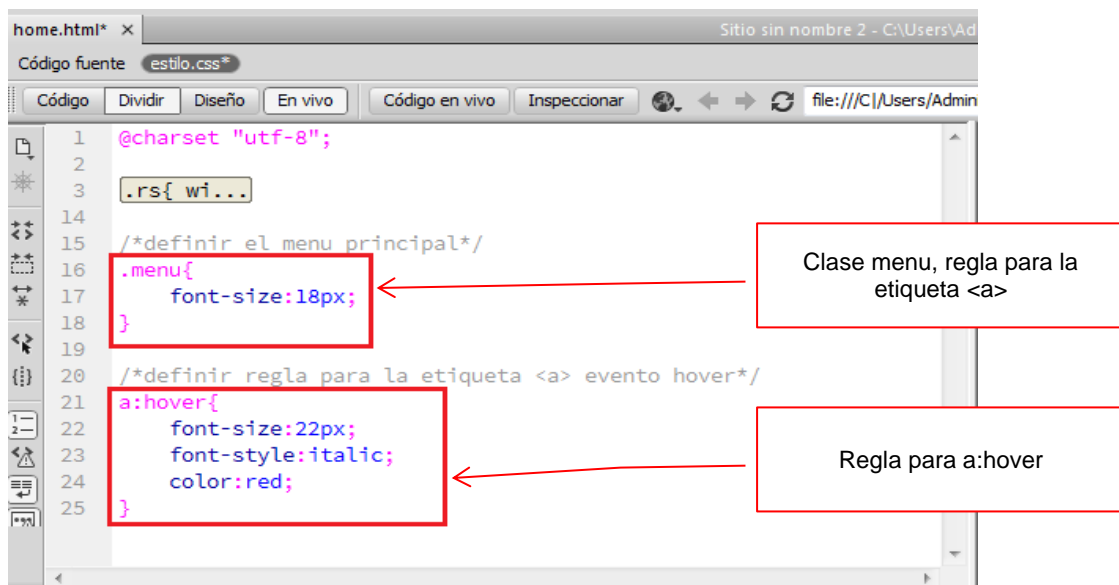


## Definiendo las reglas de los enlaces (Menu) en el archivo estilo.css

En el archivo CSS, estilo.css definimos las reglas para la etiqueta <a> y hover.

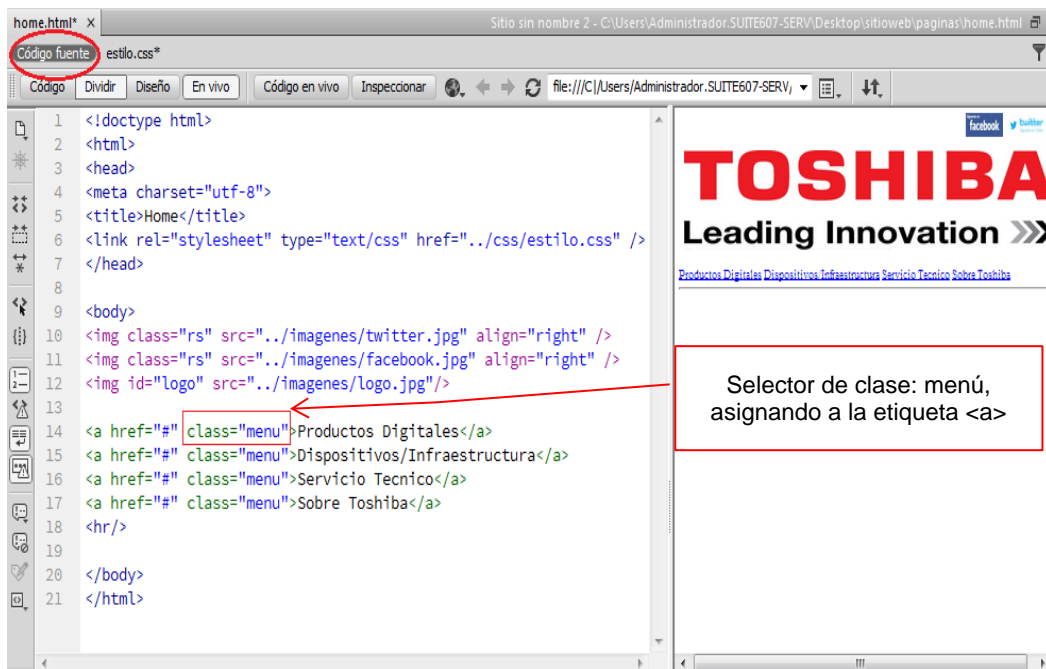
- Defina la clase **menu**, es una regla para los enlaces.
- Defina una regla hover.

Acontinuación guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <a> y sus reglas en la página

Definida las reglas en estilo.css, dentro del body defina las etiquetas <a> agregando a cada una su selector de clase: menú, tal como se muestra. Presiona la tecla F5 para visualizar el diseño de la página.

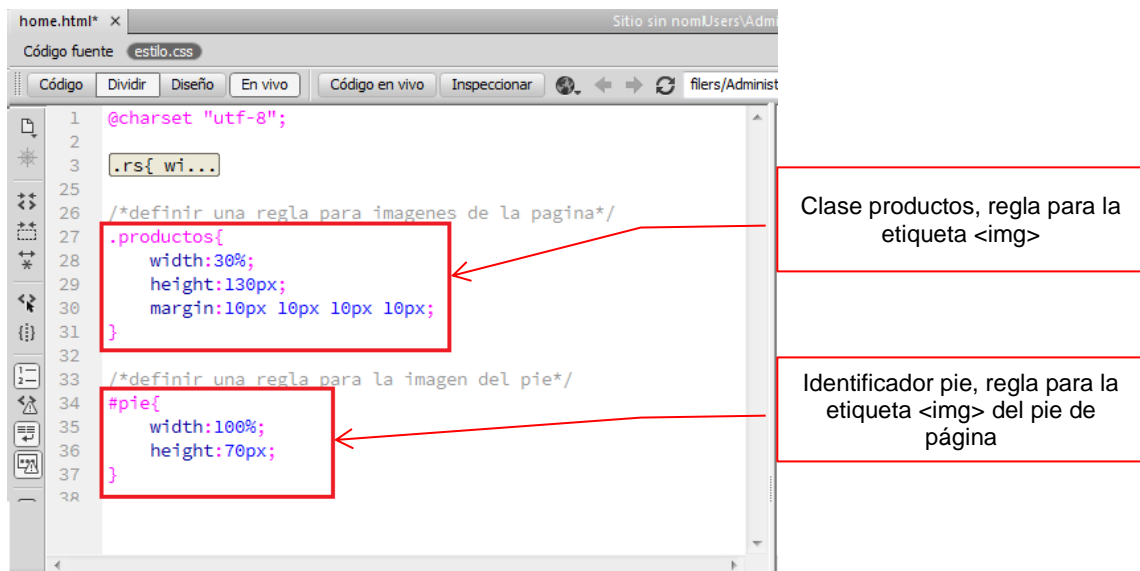


## Definiendo las reglas de las imágenes (promoción de productos y servicios y pie) en el archivo estilo.css

En el archivo CSS, estilo.css definimos las reglas para la etiqueta <img>.

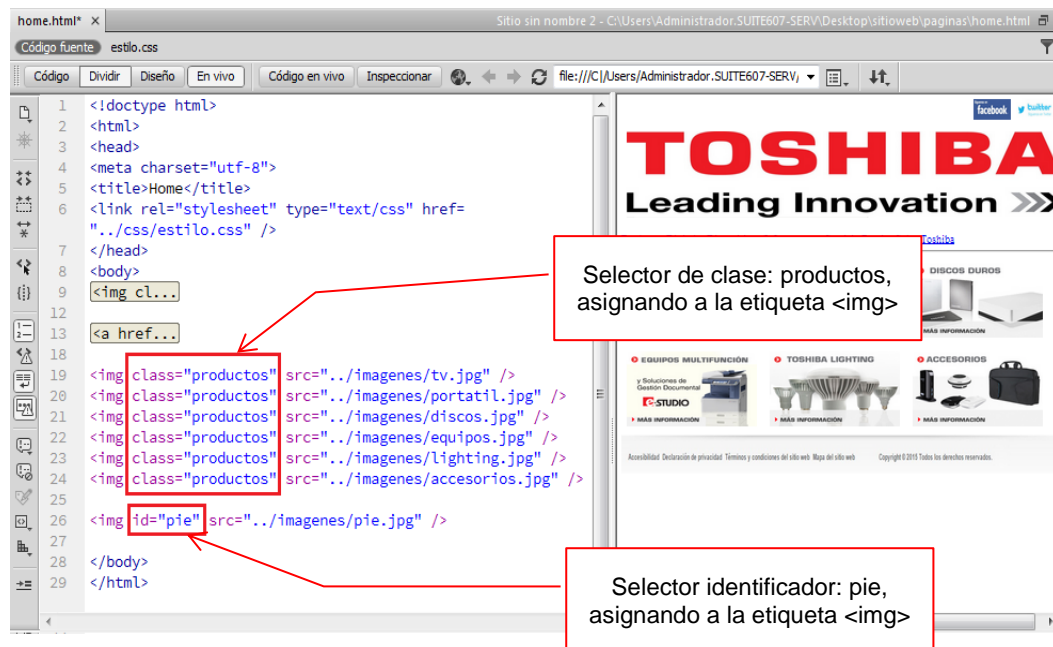
- Defina la clase **productos**, es una regla para las imágenes de promoción de productos y servicios toshiba.
- Defina el identificador **pie**, para la imagen del pie

A continuación, guardar el archivo estilo.css: CTRL+ S



## Agregando las etiquetas <img> para el cuerpo y pie en la página

Definida las reglas en estilo.css, dentro del body, defina las etiquetas <img> para promocionar los productos agregando la clase **productos**; y la etiqueta <img> para el pie de página agregando su identificador **pie**, tal como se muestra. Presiona la tecla F5 para visualizar el diseño.



Para ejecutar a través del navegador, primero guardar la página: CTRL+S y presiona la tecla F12 para ejecutar la página con el navegador.

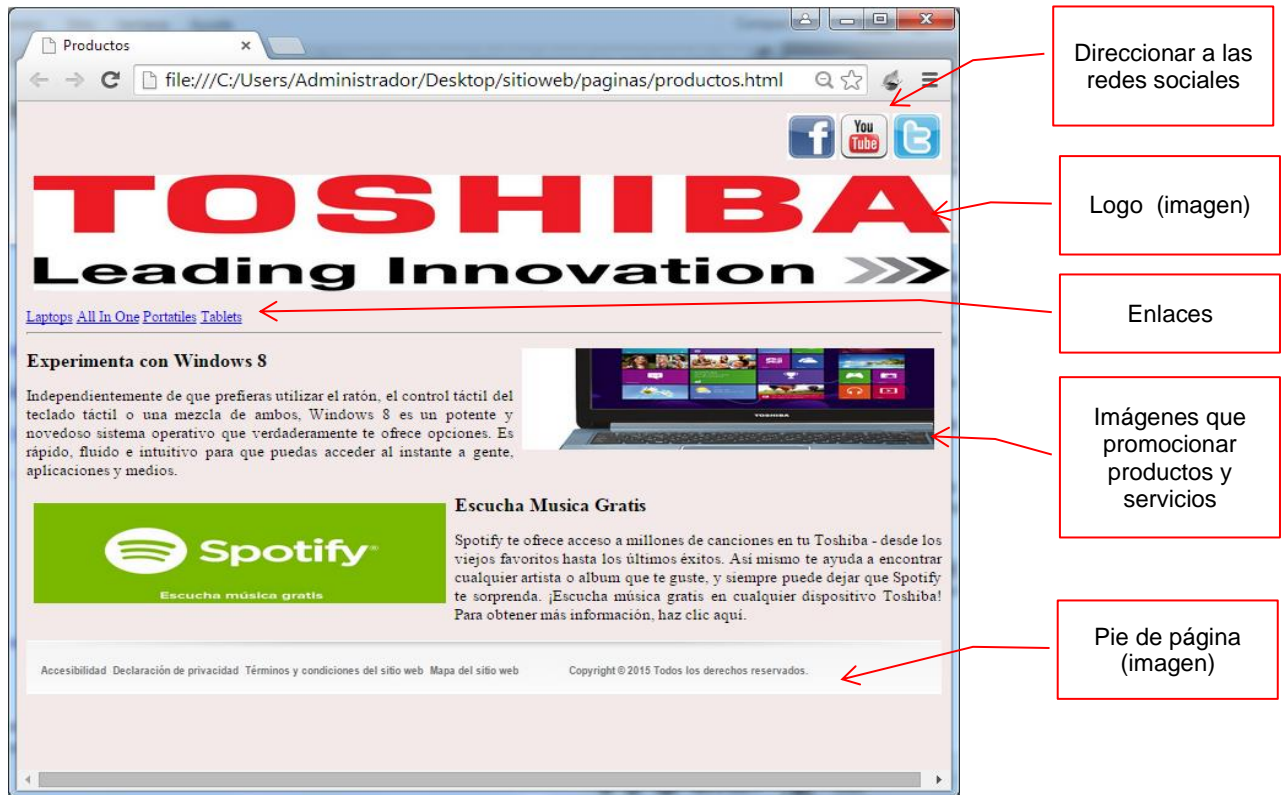




## LABORATORIO 4.2

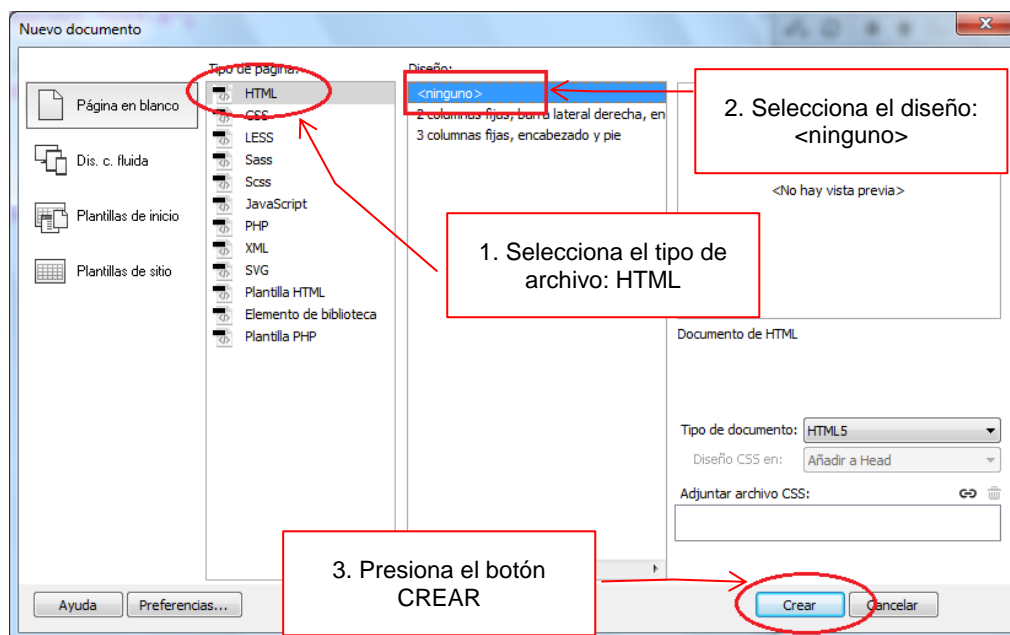
### Trabajando con Estilos CSS

Se nos pide implementar una página para promocionar los productos de la empresa Toshiba, tal como se visualiza.



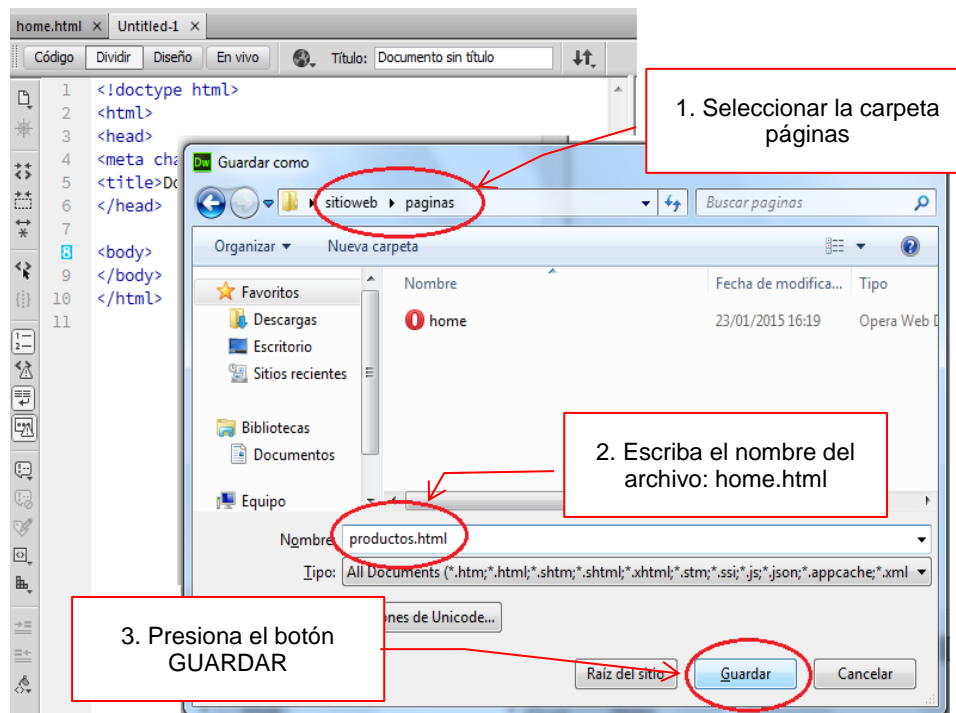
### Agregando la pagina HTML

A continuación, creamos un archivo HTML, el cual no tendrá diseño (ninguno), tal como se muestra.



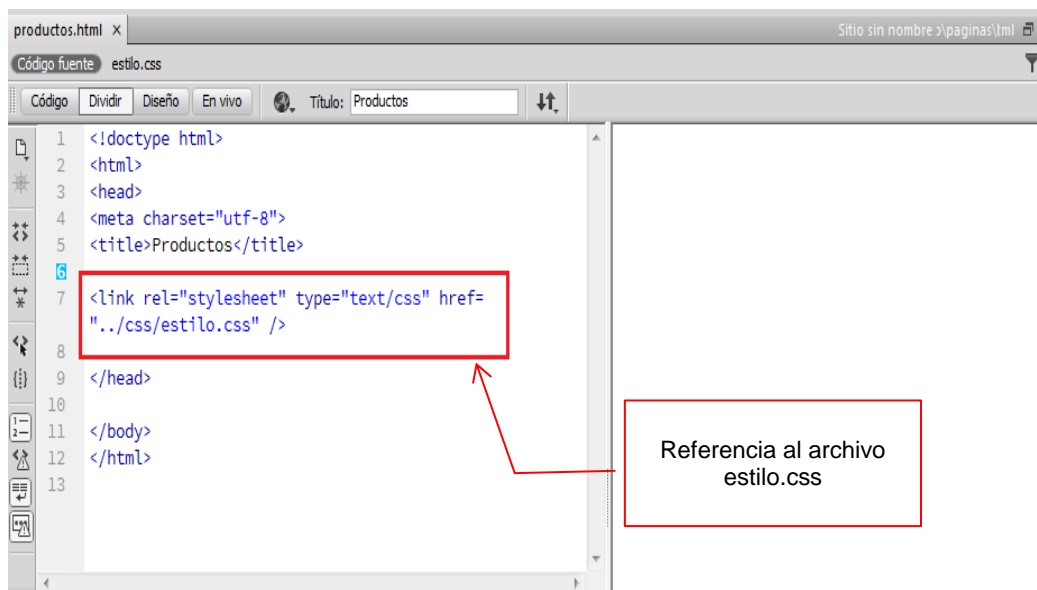


Creado el documento HTML, procedemos a guardarla, presiona la combinación CTRL+S



### Enlazando a la hoja de estilo

Creada la página, definimos, dentro del <head> la etiqueta <link> el cual hace una referencia al archivo CSS estilo.css, tal como se muestra.

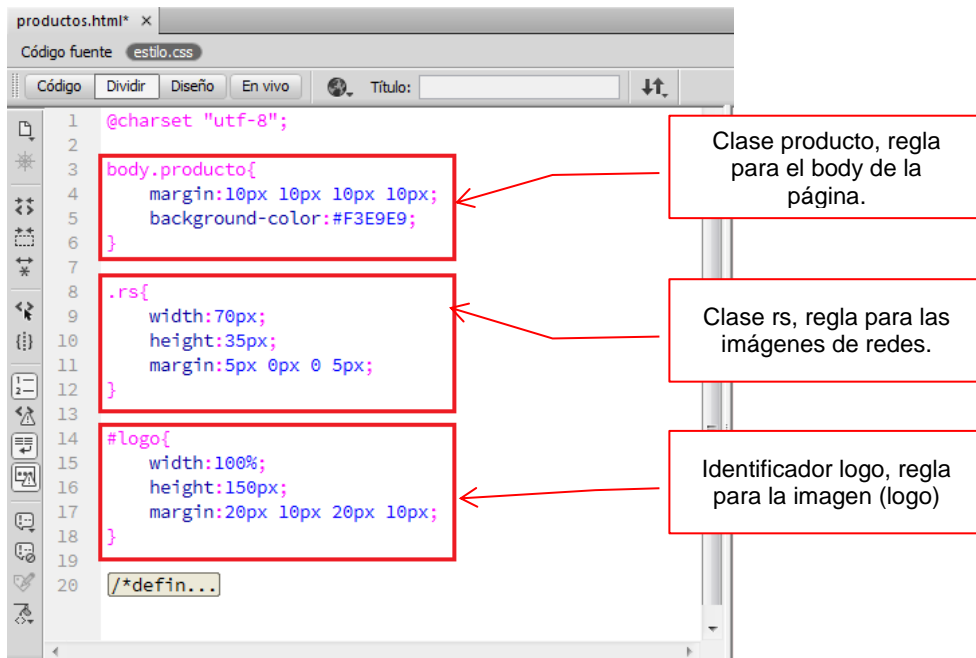


## Definiendo reglas al body y cabecera de la pagina

En el archivo CSS, estilo.css definimos las primeras dos reglas: .rs y #logo

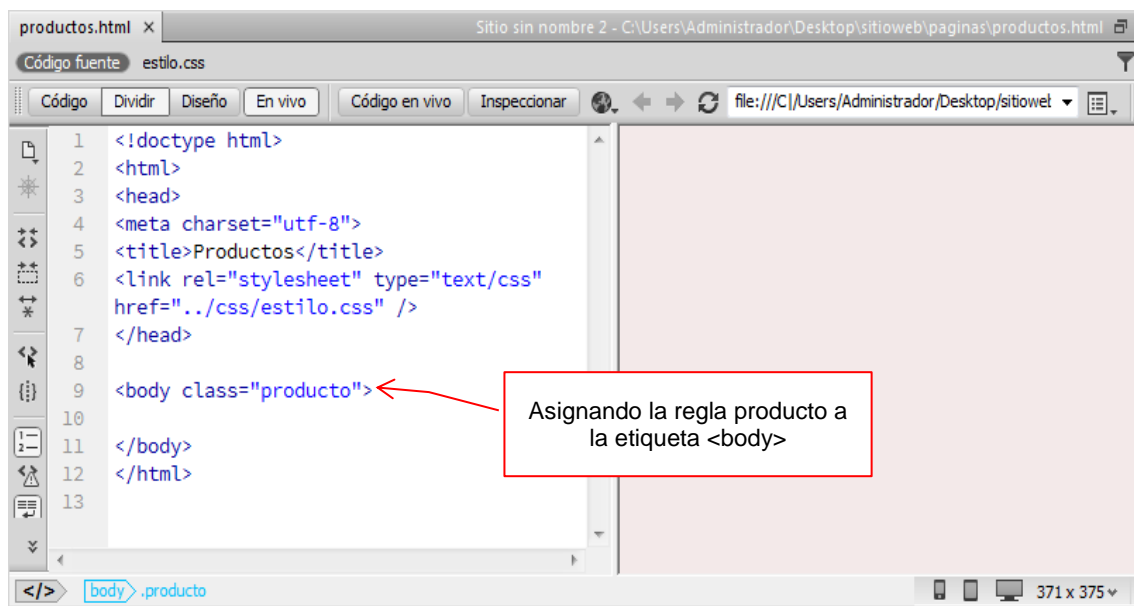
- Defina la clase **body.producto**, es una regla para el body la pagina wb que vamos a diseñar.
- Defina la clase **rs**, es una regla para las imágenes de redes sociales.
- El identificador **logo**, es una regla para la imagen de logo de la pagina.

Acontinuación guardar el archivo estilo.css: CTRL+S



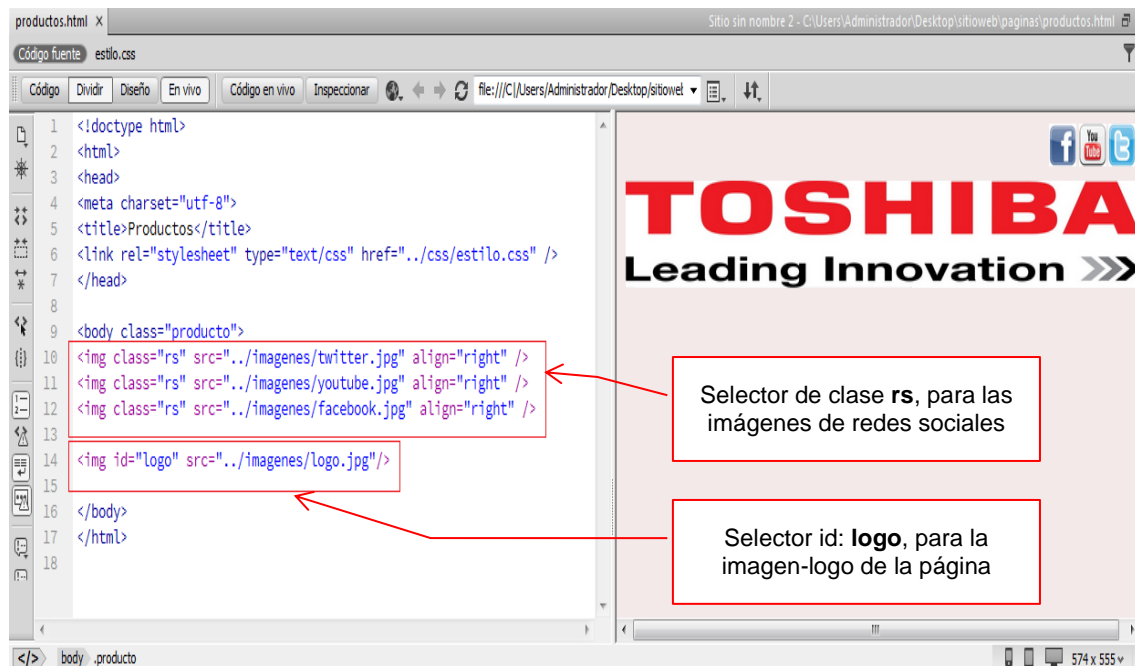
## Agregando las etiquetas y reglas en la página

Definida las reglas en estilo.css, asigne la regla producto a la etiqueta body, para asignar color de fondo a la pagina.



## Agregando las etiquetas y reglas en la página

Definida las reglas en estilo.css, dentro del body defina las etiquetas <img> agregando a cada una su selector: de clase para las redes sociales y de identificador para el logo de la pagina.

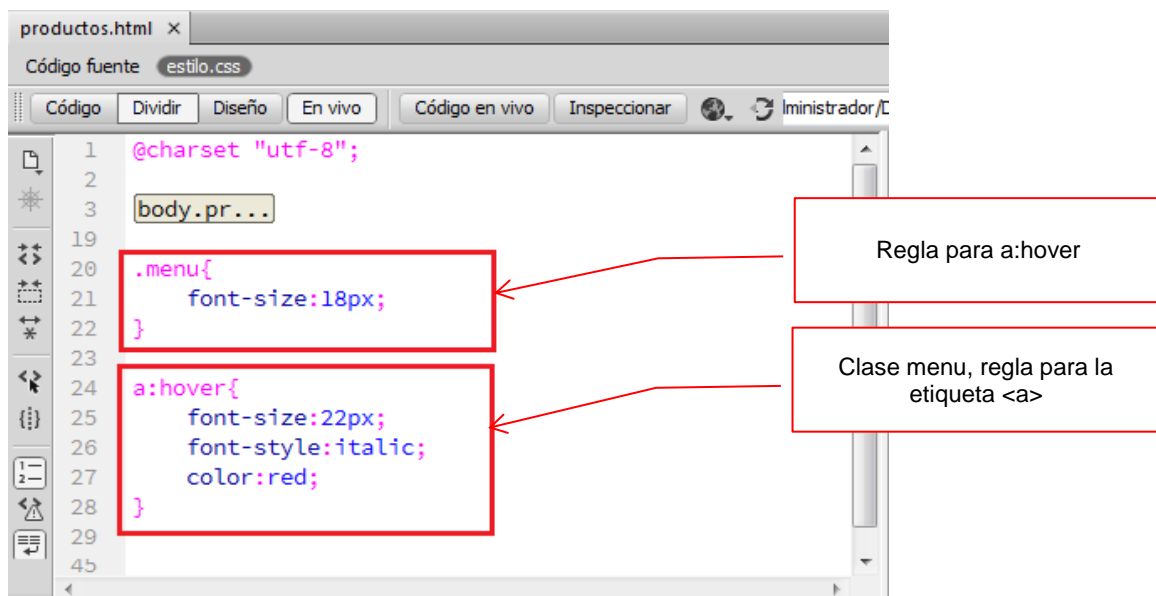


## Definiendo las reglas de los enlaces (Menu) en el archivo estilo.css

En el archivo CSS, estilo.css definimos las reglas para la etiqueta <a> y hover.

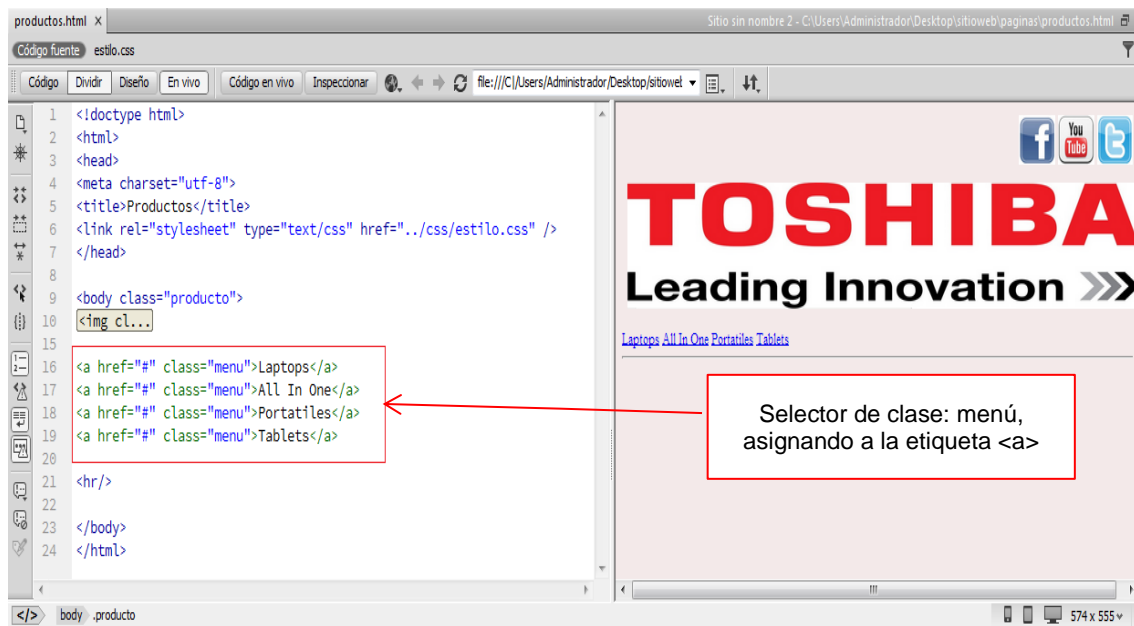
- Defina la clase **menu**, es una regla para los enlaces.
- Defina una regla **hover**.

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <a> y sus reglas en la página

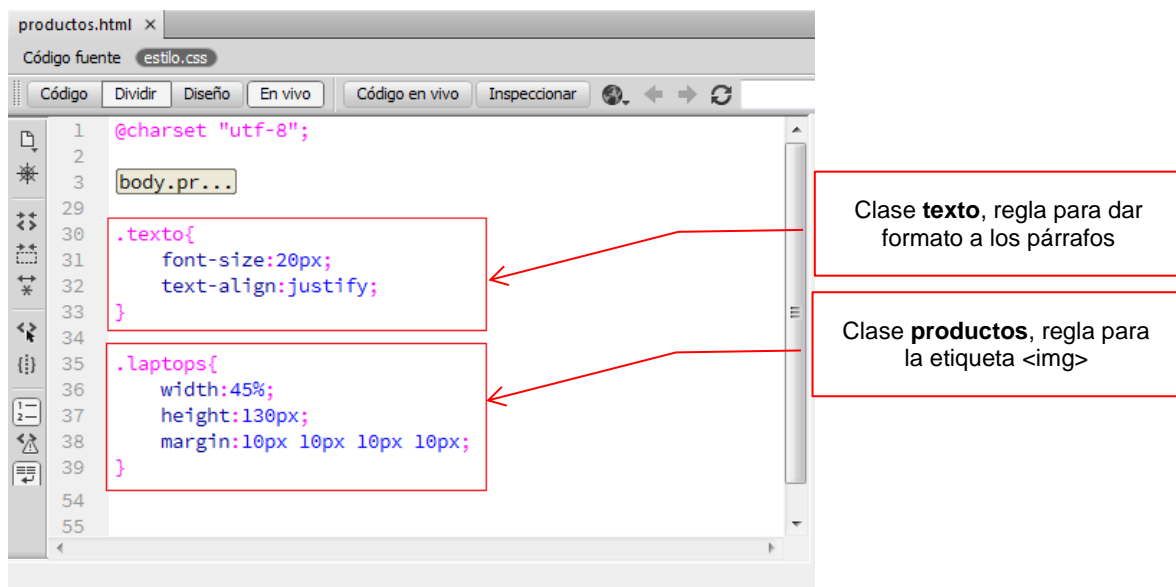
Definida las reglas en estilo.css, dentro del body defina las etiquetas <a> agregando a cada una su selector de clase: menú, tal como se muestra. Presiona la tecla F5 para visualizar el diseño de la página



## Definiendo las reglas de los párrafos y de las imágenes en estilo.css

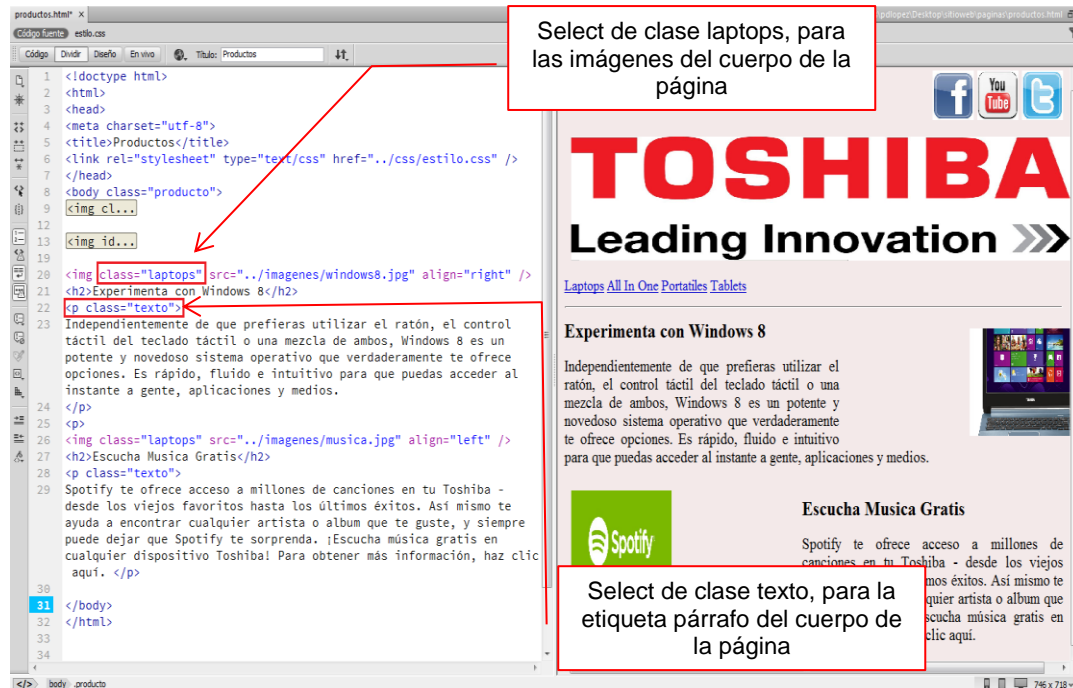
En el archivo CSS, estilo.css definimos las reglas para el párrafo y las imágenes del cuerpo

- Defina la clase **texto**, regla que da formato a los párrafos de la pagina
- Defina la clase **productos**, es una regla para las imágenes de promoción de productos y servicios toshiba.
- Acontinuación guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <p> e <img> para el cuerpo de la página

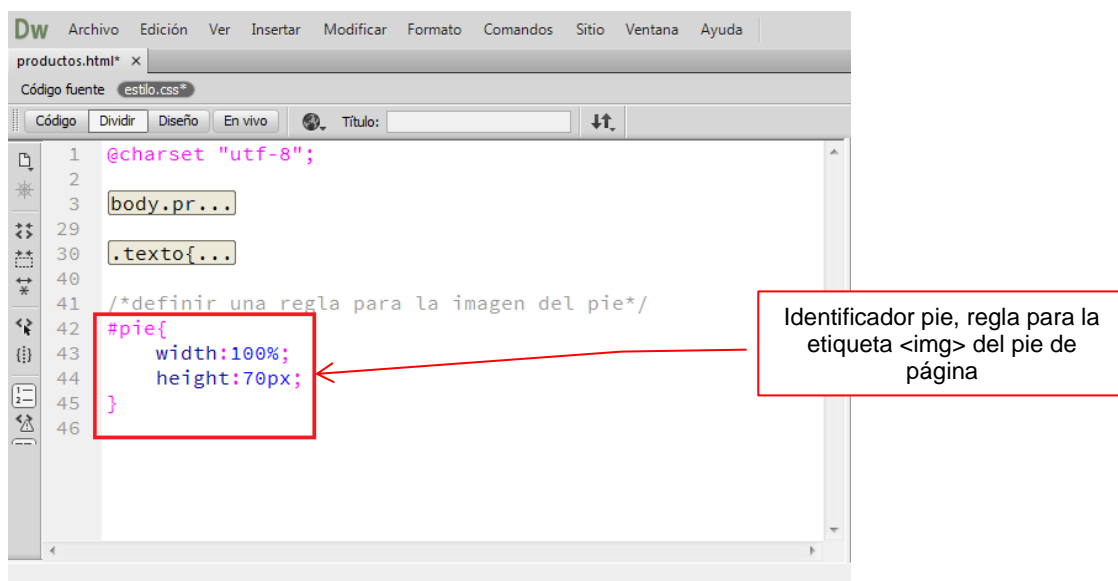
Definida las reglas en estilo.css, defina las etiquetas <p> e <img> para promocionar los productos. Utilice la clase **texto** para la etiqueta <p> y la clase **laptops** en la etiqueta <img> del cuerpo de la página. Presiona la tecla F5 para visualizar el diseño.



## Definiendo las reglas de las imágenes: pie, en el archivo estilo.css

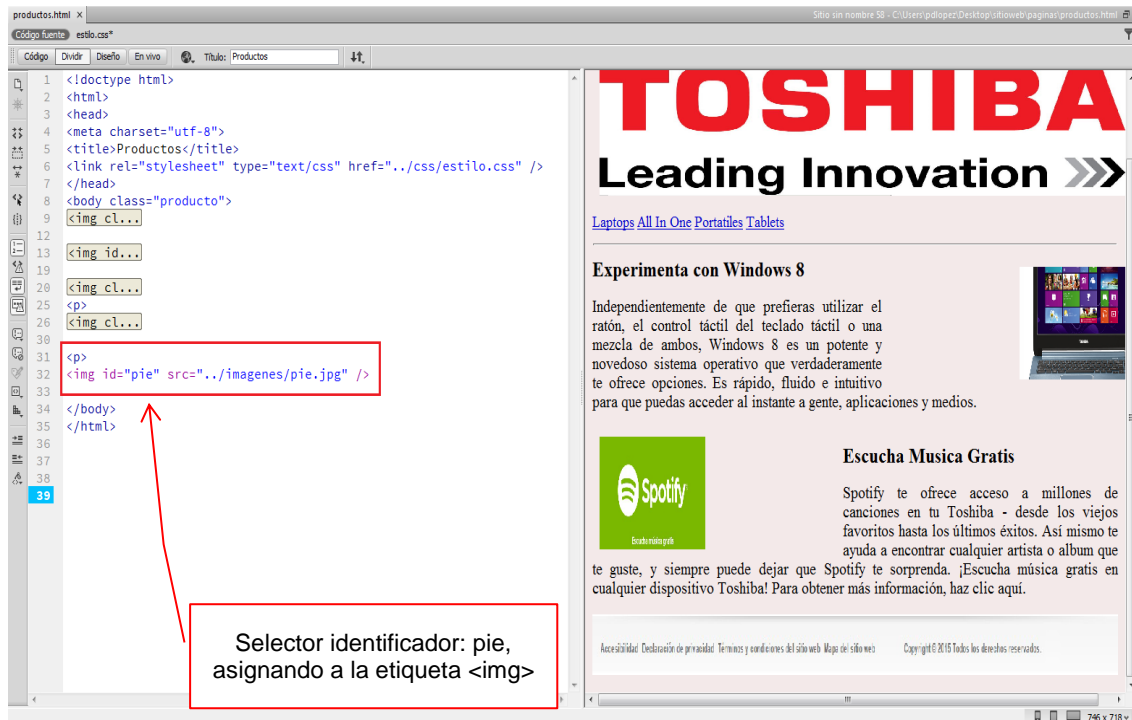
En el archivo CSS, estilo.css definimos las reglas para la etiqueta <img>.

- Defina el identificador **pie**, para la imagen del pie
- Acontinuación guardar el archivo estilo.css: CTRL+ S



## Agregando las etiquetas <img> para el pie en la página

Definida las reglas en estilo.css, dentro del body, defina las etiquetas <img> para el pie de página agregando su identificador **pie**, tal como se muestra. Presiona la tecla F5 para visualizar el diseño.



Para ejecutar a través del navegador, primero guardar la página: CTRL+S y presiona la tecla F12 para ejecutar la página con el navegador.



# Resumen

- 📖 CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, las cuales están escritas con código HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.
- 📖 El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.
- 📖 CSS funciona en base a declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML. La regla tiene dos partes: un selector y la declaración.
- 📖 Estilo interno, si deseamos crear definiciones de estilo solamente para una página del sitio web debemos crear un estilo interno. Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta **<style>** de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección **<head>**).
- 📖 Estilo externo, si deseamos que el sitio Web tenga un aspecto uniforme debemos utilizar una hoja de estilos externa y vinculada a cada documento. Las hojas de estilo se pueden almacenar en **archivos externos** y vincularlas a uno o más documentos.
- 📖 Para incluir en la página HTML los estilos definidos en archivos CSS externos se utiliza una regla especial de tipo **@import**. Las reglas de tipo **@import** siempre preceden a cualquier otra regla CSS (con la única excepción de la regla **@charset**).
- 📖 El selector de tipo o etiqueta, es un tipo de selector identifica a un tipo de elemento que conforma el código HTML modificando el aspecto predeterminado de la etiqueta. Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres **<** y **>**) correspondiente a los elementos que se quieren seleccionar.
- 📖 El selector de clase es cuando se declara una regla en la hoja de estilo, esta regla se aplica a todos los elementos HTML descritos por su selector.
- 📖 EL selector de ID, en ocasiones, es necesario aplicar estilos CSS a un único elemento de la página HTML. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 [http://librosweb.es/css\\_avanzado/capitulo\\_3.html](http://librosweb.es/css_avanzado/capitulo_3.html)

🔗 [http://www.mclibre.org/consultar/htmlcss/css/css\\_pseudoclasses.html](http://www.mclibre.org/consultar/htmlcss/css/css_pseudoclasses.html)

🔗 <http://www.sidar.org/recur/desdi/traduc/es/css/selector.html>

🔗 <http://html.conclase.net/w3c/css1-es.html>

🔗 <http://tejedoresdelweb.com/w/CSS>

🔗 <https://kechy.files.wordpress.com/2012/11/manual-css-hojas-estilo.pdf>

🔗 [http://librosweb.es/css/capitulo\\_2/selectores\\_basicos.html](http://librosweb.es/css/capitulo_2/selectores_basicos.html)







# ESTRUCTURAS Y LAYOUT

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3 y estructuras.

## TEMARIO

### 3.1 Tema 5: Creacion de Bloques

3.1.1 Etiquetas de bloques: header, nav, aside, section, article, footer, div.

3.1.1.1 Estilos básicos del bloque: width, height, background, border, margin.

3.1.1.2 Estilos para bloques: padding, background-image, border-radius, float, clear

3.1.1.3 Propiedad box-sizing

### 3.2 Tema 6: Creacion de Menú

3.2.1 Estilos a vínculos: creacion de menú

3.2.1.1 Ejercicio Integrador

## ACTIVIDADES PROPUESTAS

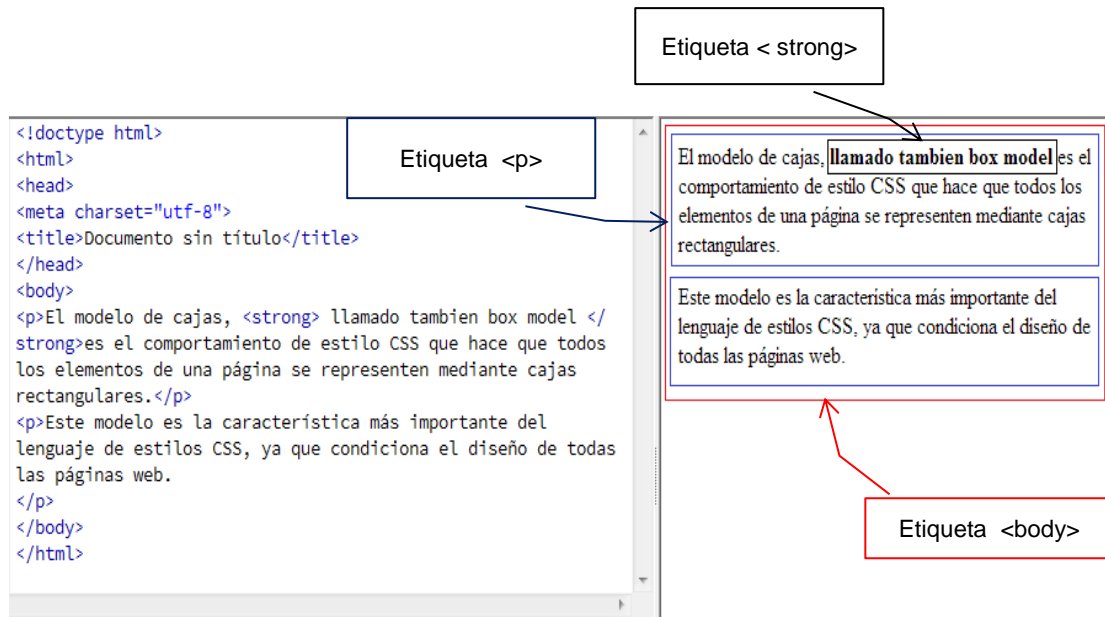
- Los alumnos diseñan páginas web con etiquetas HTML5, hojas de estilo CSS3 y bloques de contenidos
- Los alumnos diseñan un menú en una página web



### 3.1 CREACION DE BLOQUES

El modelo de cajas es el comportamiento de estilo CSS que hace que todos los elementos de una página se representen mediante cajas rectangulares. Este modelo es la característica más importante del lenguaje de estilos CSS, ya que condiciona el diseño de todas las páginas web.

Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento. La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:



#### 3.1.1 Etiquetas de bloques: header, nav, aside, section, article, footer.

Piensa en una página Web como un documento. Los documentos tienen encabezados, pies de página, capítulos y otras convenciones diferentes que dividen el documento en partes lógicas. La razón para trabajar con etiquetas estructurales es dividir una página Web en partes lógicas que describan el tipo de contenido que incluyen.

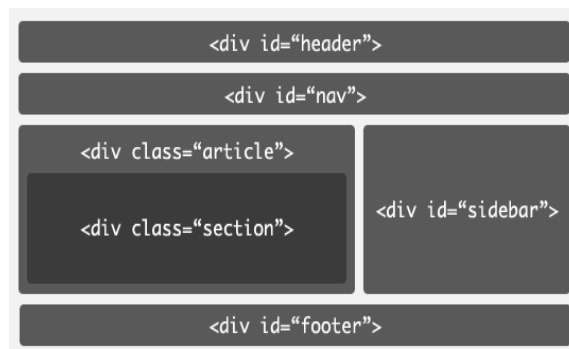
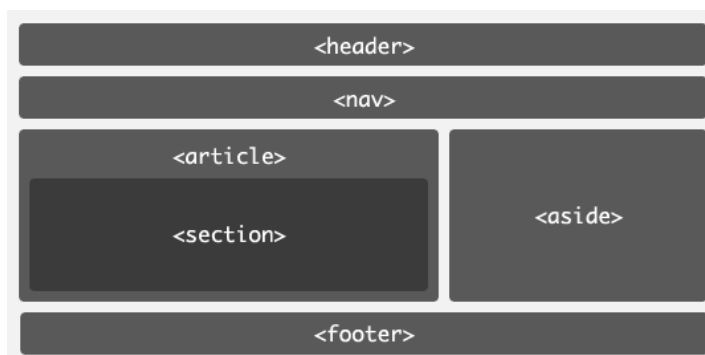


Figura 1

Referencia: <http://www.ticweb.es/html5-las-nuevas-etiquetas-y-las-nuevas-posibilidades-de-este-lenguaje/>

HTML 4 tiene elementos semánticos que permiten definir con claridad las diferentes partes de una página web, como formularios, listas, párrafos, tablas, etc.; pero estos elementos dependen mucho de elementos **<div>** y **<span>** con diferentes identificadores y atributos de clase para definir varias características, tales como: menú de navegación: `<div id="menú">`, encabezados `<div id="cabecera">`, pie de página `<div id="pie">`, etc.



**Figura 2**

Referencia: <http://www.ticweb.es/html5-las-nuevas-etiquetas-y-las-nuevas-posibilidades-de-este-lenguaje/>

HTML5 nos aporta una serie de etiquetas nuevas que permiten mejorar la semántica de nuestra página. Los elementos de HTML5 nuevos que vamos a cubrir en este artículo son:

Elemento	Descripción
<code>&lt;header&gt;</code> (cabecera)	Se utiliza para contener la cabecera del sitio.
<code>&lt;footer&gt;</code> (pie de página)	Contiene el pie de página de un sitio.
<code>&lt;nav&gt;</code>	Contiene la funcionalidad de navegación para la página
<code>&lt;article&gt;</code> (artículo)	Contiene una pieza independiente de contenido.
<code>&lt;section&gt;</code> (sección)	Se utiliza bien para agrupar artículos en diferentes temas, bien para definir las diferentes secciones de un solo artículo.
<code>&lt;aside&gt;</code> (barra lateral)	Define un bloque de contenido relacionado con el contenido principal que lo rodea.
<code>&lt;hgroup&gt;</code> (heading group, o grupo de encabezados)	Se utiliza para incluir más de un encabezado si quieres que cuente como un único encabezado en la estructura de encabezado de la página.
<code>&lt;figure&gt;</code> y <code>&lt;figcaption&gt;</code> (figura y cita de figura)	Se usan para encapsular una figura como un único elemento y contener el título de la figura, respectivamente.

### **<header>**

La etiqueta `<header>` es un contenedor que forma la cabecera de la página, generalmente con un logotipo o gráfico de la compañía, el título de página principal, etc.

`<header>` puede ser utilizado varias veces en un documento y por lo general contiene al menos un elemento `<h1>`-`<h6>`

### <hgroup>

Notarás que en el código anterior el único contenido de la cabecera es un elemento <hgroup>, que rodea dos encabezados. Lo que pretendo es especificar el encabezado de nivel superior del documento, más un subtítulo o 'tagline'. Quiero que sólo cuente el nivel superior de encabezado en la jerarquía de encabezados del documento y eso es exactamente lo que hace <hgroup>: que un grupo de encabezados cuente como un único encabezado a efectos de la estructura del documento.

### <nav>

La etiqueta <nav> es utilizado para demarcar una sección que contiene los enlaces a otros contenidos o secciones. Los enlaces se pueden anidar dentro de una lista, como una lista desordenada <ul>

### <section>

La etiqueta <section> se usa para representar un bloque de nuestra página que tiene valor semántico, es decir, que aporta un significado a la página. Realmente si tenemos que clasificar por la importancia del significado, el ranking sería: article sería la etiqueta que contiene la información más relevante, section la que contiene información menos relevante, y div que contiene información que no aporta significado ninguno.

### <article>

La etiqueta <article> se utiliza para contenido independientes que tendrían sentido fuera del contexto de la página actual, y podrían sindicarse perfectamente. Eso incluiría las entradas de un blog, un vídeo con su transcripción, una noticia, o una parte de una historia por capítulos.

### <footer>

La etiqueta <footer> se utiliza para contener el contenido de pie de página del sitio. El pie de página se utiliza para poner: avisos de copyright, datos de contacto a las declaraciones de accesibilidad, información de licencias y varios otros enlaces secundarios.

#### 3.1.1.1 Estilos básicos del bloque: width, height, background, border, margin.

Las partes que componen un bloque o caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

- Contenido (**content**): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- Ancho (**width**): controla el ancho de la caja
- Height (alto): controla la altura de la caja
- Borde (**border**): línea que encierra completamente el contenido y su relleno.
- Margen (**margin**): separación opcional existente entre la caja y el resto de cajas adyacentes.

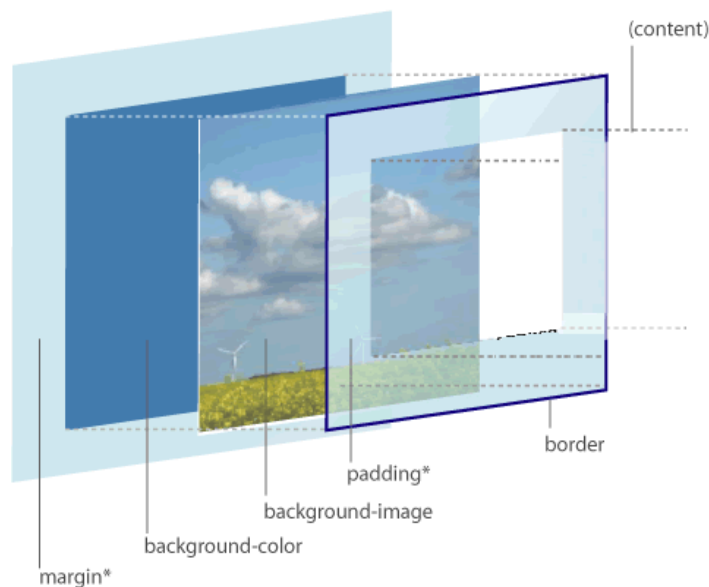


Figura 3

Referencia: [http://librosweb.es/css/capitulo\\_4.html](http://librosweb.es/css/capitulo_4.html)

La propiedad **width** es una propiedad CSS que controla la anchura de la caja de los elementos. Esta propiedad no admite valores negativos y los valores en porcentaje se calculan a partir de la anchura de su elemento padre. El valor **inherit** indica que la anchura del elemento se hereda de su elemento padre. El valor **auto**, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad.

La propiedad **height** es una propiedad CSS que controla la altura de los elementos. Al igual que sucede con width, esta propiedad no admite valores negativos. Si se indica un porcentaje, se toma como referencia la altura del elemento padre. Si el elemento padre no tiene una altura definida explícitamente, se asigna el valor **auto**. El valor **inherit** indica que la altura del elemento se hereda de su elemento padre.

La propiedad margin define 4 propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento: **margin-top**, **margin-right**, **margin-bottom** y **margin-left**

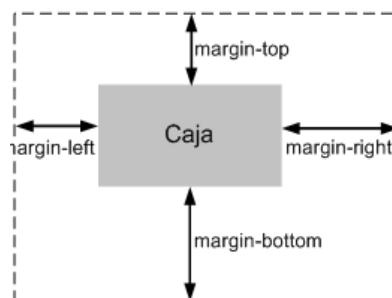


Figura 4

Referencia: [http://librosweb.es/css/capitulo\\_4.html](http://librosweb.es/css/capitulo_4.html)

La notación de la propiedad **margin** admite entre uno y cuatro valores, con el siguiente significado:

- Si solo se indica un valor, todos los márgenes tienen ese valor.
- Si se indican dos valores, el primero se asigna al margen superior e inferior y el segundo se asigna a los márgenes izquierdo y derecho.
- Si se indican tres valores, el primero se asigna al margen superior, el tercero se asigna al margen inferior y el segundo valor se asigna al margen izquierdo y derecho.
- Si se indican los cuatro valores, el orden de asignación es: margen superior, margen derecho, margen inferior y margen izquierdo.

La propiedad **border** permite modificar el aspecto de cada uno de los cuatro bordes de la caja de un elemento. Para cada borde se puede establecer su anchura o grosor, su color y su estilo, por lo que en total CSS define 20 propiedades relacionadas con los bordes.

La anchura se controla con las cuatro propiedades siguientes: **border-top-width**, **border-right-width**, **border-bottom-width** y **border-left-width**. Si se quiere establecer de forma simultánea el ancho de todos los bordes de una caja, utilizamos la propiedad **border-width**.

El color de los bordes se controla con las cuatro propiedades siguientes: **border-top-color**, **border-right-color**, **border-bottom-color** y **border-left-color**. Si se quiere establecer de forma simultánea el color de todos los bordes de una caja utilizamos la propiedad **border-color**.

El estilo de los bordes se controla con las cuatro propiedades siguientes: **border-top-style**, **border-right-style**, **border-bottom-style** y **border-left-style**. Los bordes más utilizados son **solid** y **dashed**, seguidos de **double** y **dotted**. Si se quiere establecer de forma simultánea el color de todos los bordes de una caja utilizamos la propiedad **border-style**.

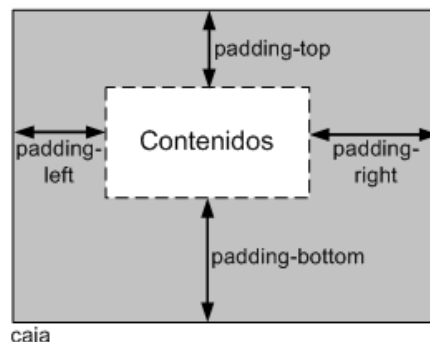
### 3.1.1.2 Estilos para bloques: **padding**, **background-image**, **border-radius**, **float**, **clear**

Las nuevas propiedades CSS3 son extremadamente poderosas y deben ser estudiadas una por una, pero para facilitar su aprendizaje vamos a aplicar todas ellas sobre la misma plantilla

- Relleno (**padding**): espacio libre opcional existente entre el contenido y el borde.
- Imagen de fondo (**background image**): imagen que se muestra por detrás del contenido y el espacio de relleno.
- Color de fondo (**background color**): color que se muestra por detrás del contenido y el espacio de relleno.
- Borde de radio (**border-radius**): efecto de las esquinas redondeadas
- Posicionamiento (**float**): establece posicionamiento flotante de la caja.
- Clear: modifica el comportamiento de las cajas flotantes.

La propiedad **padding** o relleno define un espacio libre entre el contenido y el borde de la caja.

Esta propiedad define cuatro propiedades para controlar cada uno de los espacios de relleno horizontal y vertical: **padding-top**, **padding-right**, **padding-bottom**, **padding-left**.



**Figura 5**

Referencia: [http://librosweb.es/css/capitulo\\_4.html](http://librosweb.es/css/capitulo_4.html)

Si se quiere establecer de forma simultánea el relleno de una caja utilizamos la propiedad **padding**.

La propiedad **background-color** permite mostrar un color de fondo sólido en la caja de un elemento.

Para crear efectos gráficos avanzados, es necesario utilizar la propiedad **background-image**, que permite mostrar una imagen como fondo de la caja de cualquier elemento.

En ocasiones, no es conveniente que la imagen de fondo se repita horizontal y verticalmente. Para ello, CSS introduce la propiedad **background-repeat** que permite controlar la forma de repetición de las imágenes de fondo. El valor **repeat** indica que la imagen se debe repetir en todas direcciones y por tanto, es el comportamiento por defecto. El valor **no-repeat** muestra una sola vez la imagen y no se repite en ninguna dirección. El valor **repeat-x** repite la imagen sólo horizontalmente y el valor **repeat-y** repite la imagen solamente de forma vertical.

Para hacer que una imagen de fondo se muestre fija al desplazar la ventana del navegador, se debe añadir la propiedad **background-attachment: fixed**.

La propiedad **border-radius** permite que las esquinas sean bordes redondeados. Esta propiedad en este momento es experimental por lo que debemos usar los prefijos **-moz-** y **-webkit-** para que funcionen en navegadores basados en motores Gecko y WebKit, como Firefox, Safari y Google Chrome. Si todas las esquinas tienen la misma curvatura podemos utilizar un solo valor.

Para aplicarla efectivamente a sus documentos, debe declararla con los correspondientes prefijos para los navegadores más comunes son los siguientes:

- **-moz-** para Firefox.
- **-webkit-** para Safari y Chrome.
- **-o-** para Opera.



- -khtml- para Konqueror.
- -ms- para Internet Explorer.
- -chrome- específico para Google Chrome.

```
#principal {  
    display: block;  
    width: 500px;  
    margin: 50px auto;  
    padding: 15px;  
    text-align: center;  
    border: 1px solid #999999;  
    background: #DDDDDD;  
    -moz-border-radius: 20px 10px 30px 50px;  
    -webkit-border-radius: 20px 10px 30px 50px;  
    border-radius: 20px 10px 30px 50px;  
}
```

### Posicionamiento

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.

El estándar de CSS define cinco modelos diferentes para posicionar una caja:

- Posicionamiento estático: se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.
- Posicionamiento relativo: variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.
- Posicionamiento absoluto: la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.
- Posicionamiento fijo: variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- Posicionamiento flotante: se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

### Posicionamiento flotante

La mayoría de estructuras de las páginas web complejas están diseñadas con el posicionamiento flotante. Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una caja flotante, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.

La propiedad CSS que permite posicionar de forma flotante una caja se denomina **float**.

Si se indica un valor **left**, la caja se desplaza hasta el punto más a la izquierda posible en esa misma línea. El resto de elementos adyacentes se adaptan y fluyen alrededor de la caja flotante.



```
img {
  float:left;
}
```

El valor **right** tiene un funcionamiento idéntico, salvo que en este caso, la caja se desplaza hacia la derecha. El valor **none** permite anular el posicionamiento flotante de forma que el elemento se muestre en su posición original.

La propiedad **clear** permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante.

La regla CSS que se aplica al segundo párrafo del ejemplo anterior es la siguiente:

```
<p style="clear: left;">...</p>
```

## Visualización

Las propiedades **display** y **visibility** controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página. Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.

La propiedad **display** permite ocultar completamente un elemento haciendo que desaparezca de la página. Como el elemento oculto no se muestra, el resto de elementos de la página se mueven para ocupar su lugar.

Los valores más utilizados son **inline**, **block** y **none**. El valor **block** muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate. El valor **inline** visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate. El valor **none** oculta un elemento y hace que desaparezca de la página. El resto de elementos de la página se visualizan como si no existiera el elemento oculto, es decir, pueden ocupar el espacio en el que se debería visualizar el elemento.

La propiedad **visibility** permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento pero no la muestra. En este caso, el resto de elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.

Empleando el valor **hidden** es posible convertir una caja en invisible para que no muestre sus contenidos. El resto de elementos de la página se muestran como si la caja todavía fuera visible, por lo que en el lugar donde originalmente se mostraba la caja invisible, ahora se muestra un hueco vacío. El valor **collapse** sólo se puede utilizar en las filas, grupos de filas, columnas y grupos de columnas de una tabla. Su efecto es similar al de la propiedad **display**, ya que oculta completamente la fila y/o columna y se pueden mostrar otros contenidos en ese lugar. Si se utiliza el valor **collapse** sobre cualquier otro tipo de elemento, su efecto es idéntico al valor **hidden**.

La propiedad **overflow** permite controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.

Los valores de la propiedad **overflow** tienen el siguiente significado:

- **visible**: el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.
- **hidden**: el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.
- **scroll**: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de scroll que permiten visualizar el resto del contenido.
- **auto**: el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad **scroll**.

### 3.1.1.3 Propiedad **box-sizing**

La propiedad **box-sizing** nos permite cambiar cómo el espacio total ocupado por un elemento en pantalla será calculado forzando a los navegadores a incluir en el ancho original los valores de las propiedades **padding** y **border**.

Cada vez que el área total ocupada por un elemento es calculada, el navegador obtiene el valor final por medio de la siguiente fórmula:

tamaño + márgenes + márgenes internos + bordes.

Por este motivo, si declaramos la propiedad `width` igual a 100 píxeles, `margin` en 20 píxeles, `padding` en 10 píxeles y `border` en 1 píxel, el área horizontal total ocupada por el elemento será:  $100+40+20+2= 162$  píxeles.

Esto significa que cada vez que declare el ancho de un elemento con la propiedad `width`, deberá recordar que el área real para ubicar el elemento en pantalla será seguramente más grande.

Dependiendo de sus costumbres, a veces podría resultar útil forzar al navegador a incluir los valores de `padding` y `border` en el tamaño del elemento. En este caso la nueva fórmula sería simplemente: tamaño + márgenes.

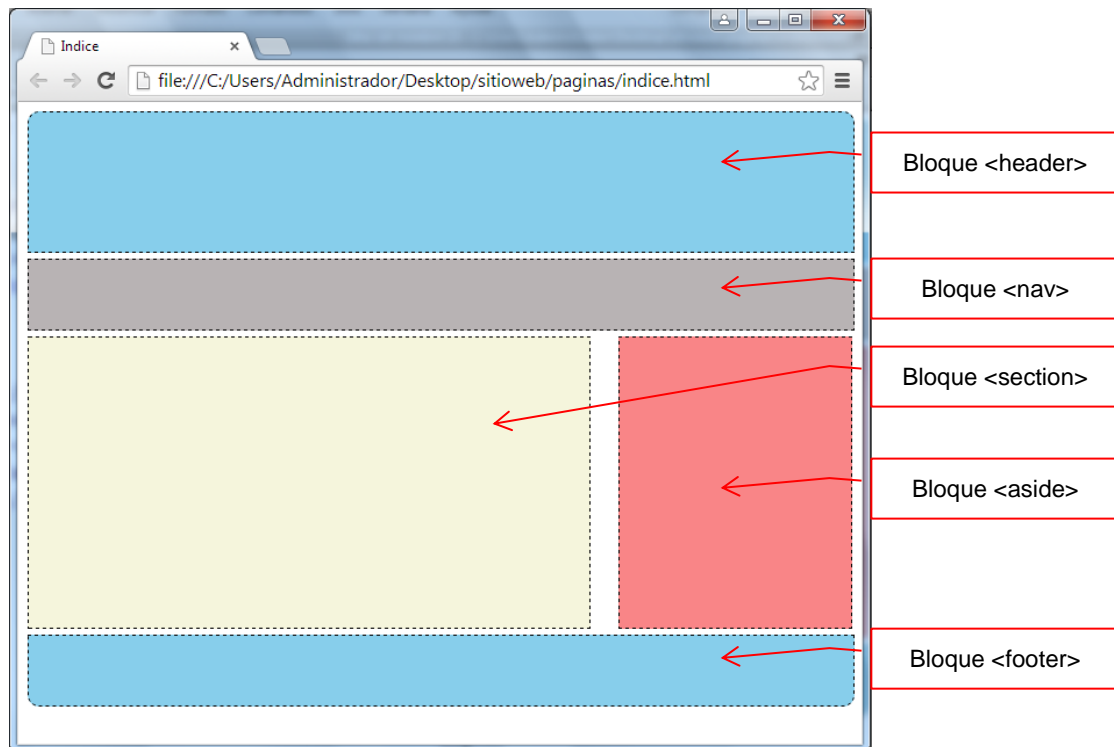
```
div {  
    width: 100px;  
    margin: 20px;  
    padding: 10px;  
    border: 1px solid;  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

La propiedad **box-sizing** puede tomar dos valores. Por defecto es configurada como **content-box**, lo que significa que los navegadores agregarán los valores de `padding` y `border` al tamaño especificado por **width** y **height**. Usando el valor **border-box** en su lugar, este comportamiento es cambiado de modo que `padding` y `border` son incluidos dentro del elemento.

## LABORATORIO 5.1

### Manejando Bloques y estilos CSS en una página HTML

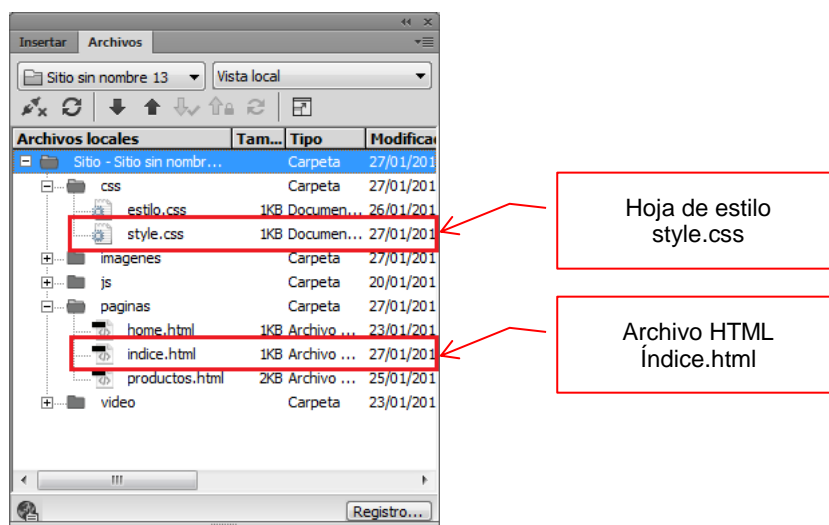
Diseña una página HTML donde visualice la estructura de bloques utilizando las etiquetas <header>, <nav>, <section>, <aside>, footer> y estilos CSS, tal como se muestra en la gráfica.



### Creando los archivos del proyecto

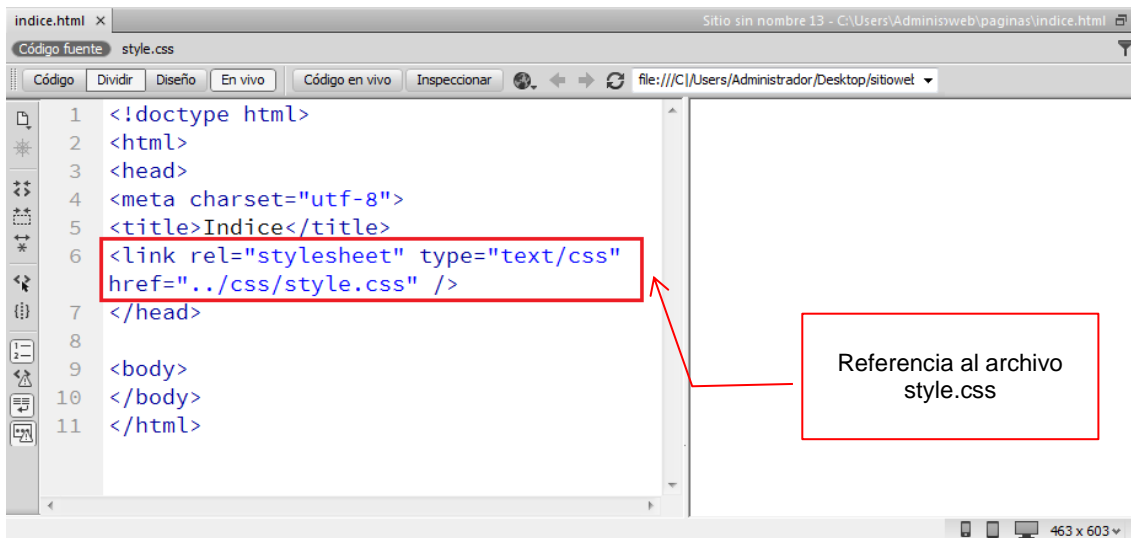
En la carpeta del sitio Web:

1. Agrega en la carpeta css, el archivo style.css
2. Agrega en la carpeta paginas, el archivo índice.html



## Enlazando a la hoja de estilo

Creada la página, definimos, dentro del <head> la etiqueta <link> el cual hace una referencia al archivo CSS style.css, tal como se muestra.

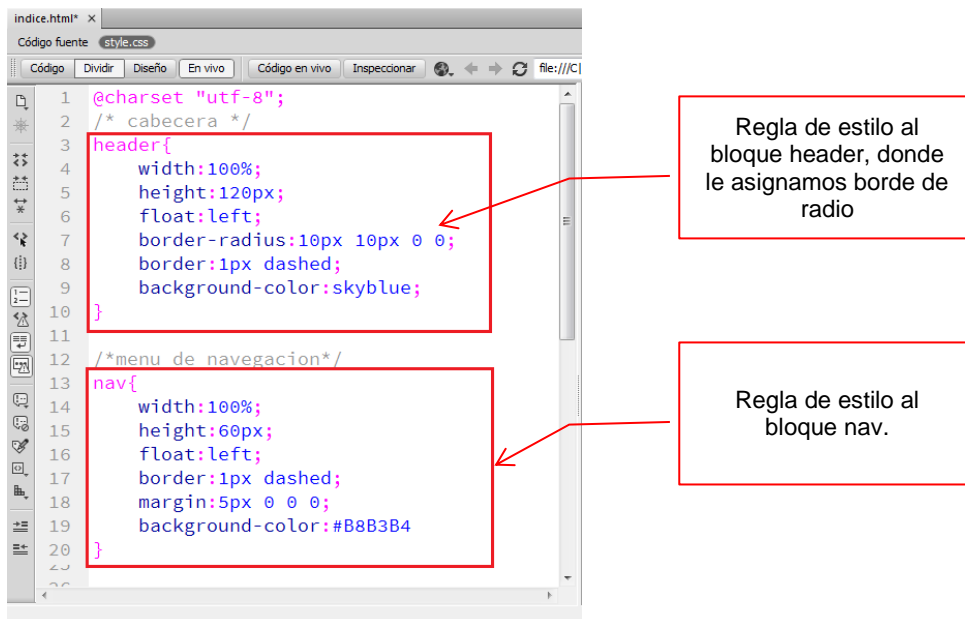


## Agregando reglas de estilo a las etiquetas <header> y <nav>

En el archivo CSS, style.css definas las reglas de estilo para <header> y <nav>:

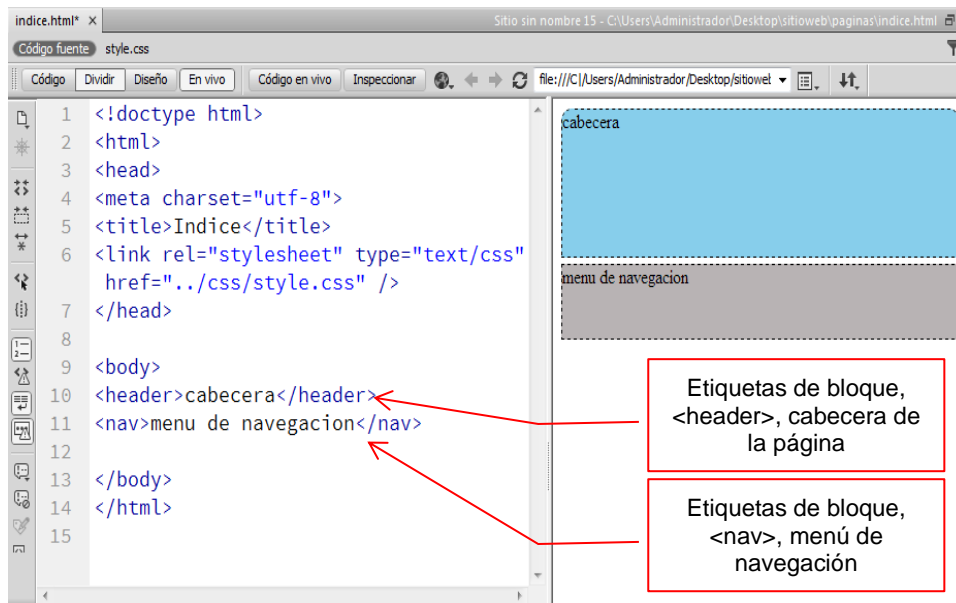
- Width: ancho del bloque
- Height: altura del bloque
- Float: bloque flotante, left, al lado izquierdo de la página
- Border: asigna borde a la caja
- Background-color: color del fondo del bloque
- Border-radius: radio de borde del bloque

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando las etiquetas de bloque en la página

Definida las reglas en style.css, dentro del body defina las etiquetas <header> y <nav>, tal como se muestra. Presiona la tecla F5 para visualizar el diseño de la página

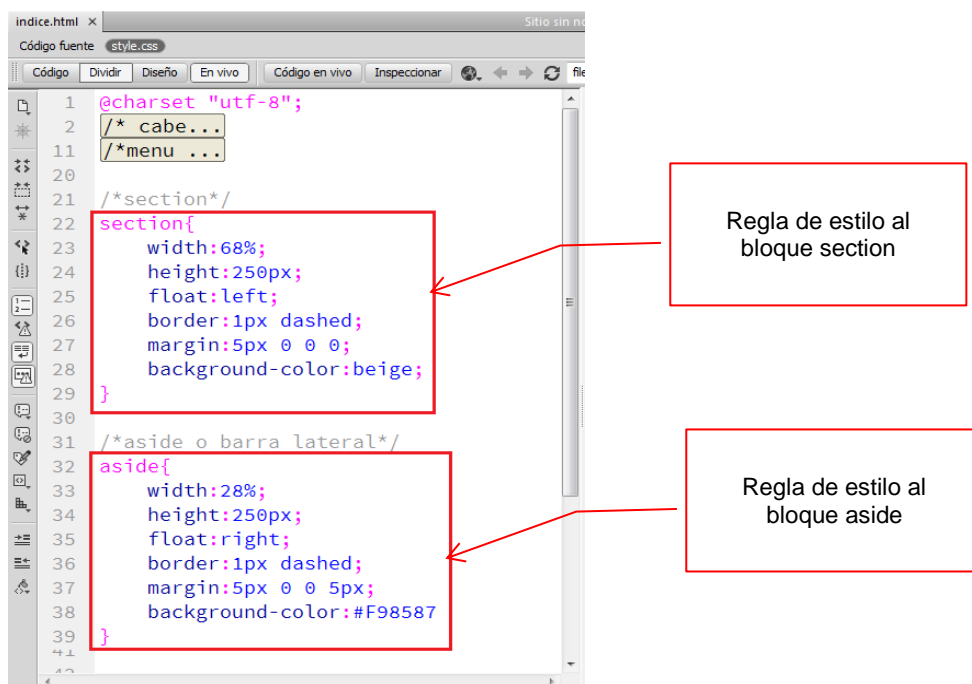


## Agregando reglas de estilo a las etiquetas <section> y <aside>

En el archivo CSS, style.css definas las reglas de estilo para <section> y <aside>:

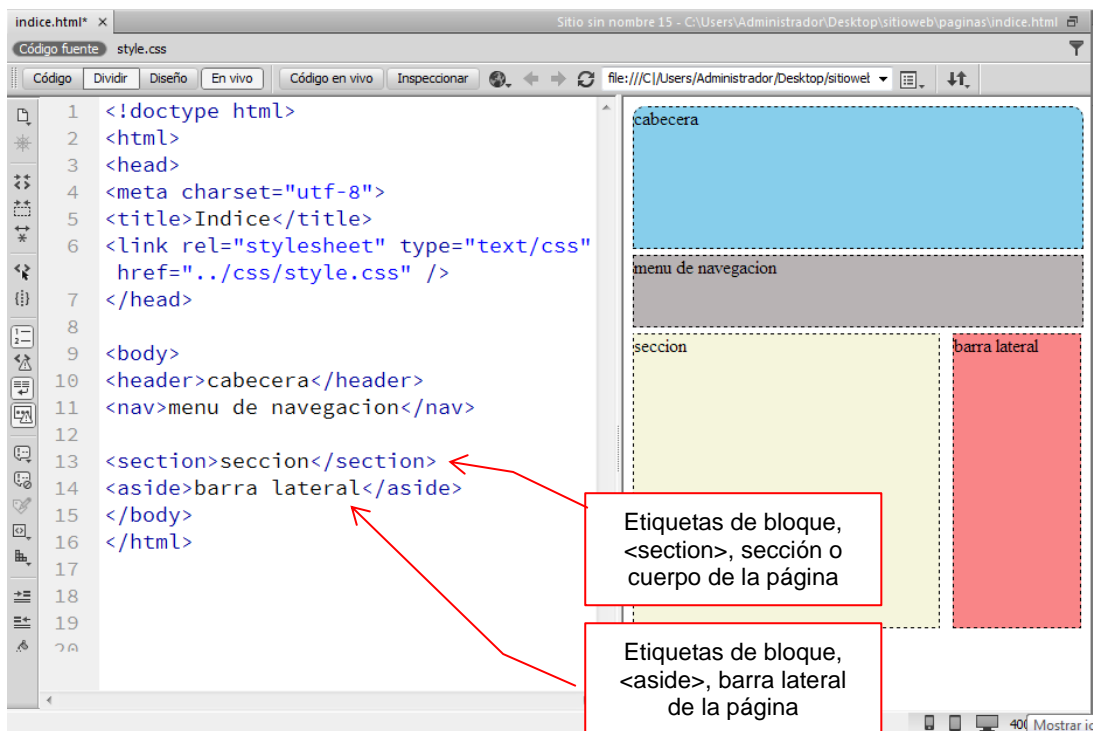
- Width y Height: ancho y altura del bloque
- Float: bloque flotante, left, al lado izquierdo y right al lado derecho de la página
- Border: asigna borde a la caja
- Background-color: color del fondo del bloque

Acontinuación guardar el archivo estilo.css: CTRL+S



## Agregando las etiquetas de bloque en la página

Definida las reglas en style.css, dentro del body defina las etiquetas <section> y <aside>, tal como se muestra. Presiona la tecla F5 para visualizar el diseño de la página.

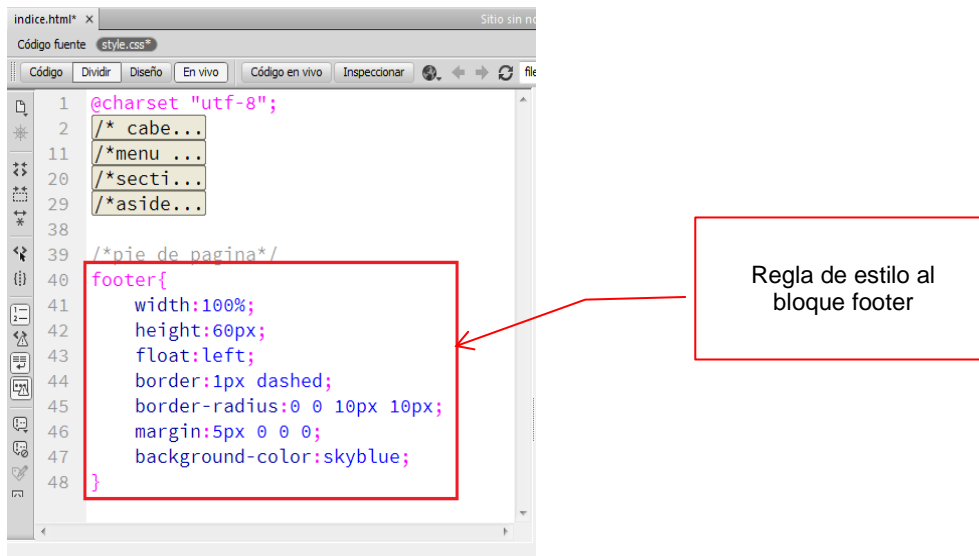


## Agregando reglas de estilo a las etiquetas <footer>

En el archivo CSS, style.css definas las reglas de estilo para <footer>

- Width y Height: ancho y altura del bloque
- Float: bloque flotante, left, al lado izquierdo de la página
- Border: asigna borde a la caja
- Background-color: color del fondo del bloque
- Border-radius: radio de borde del bloque

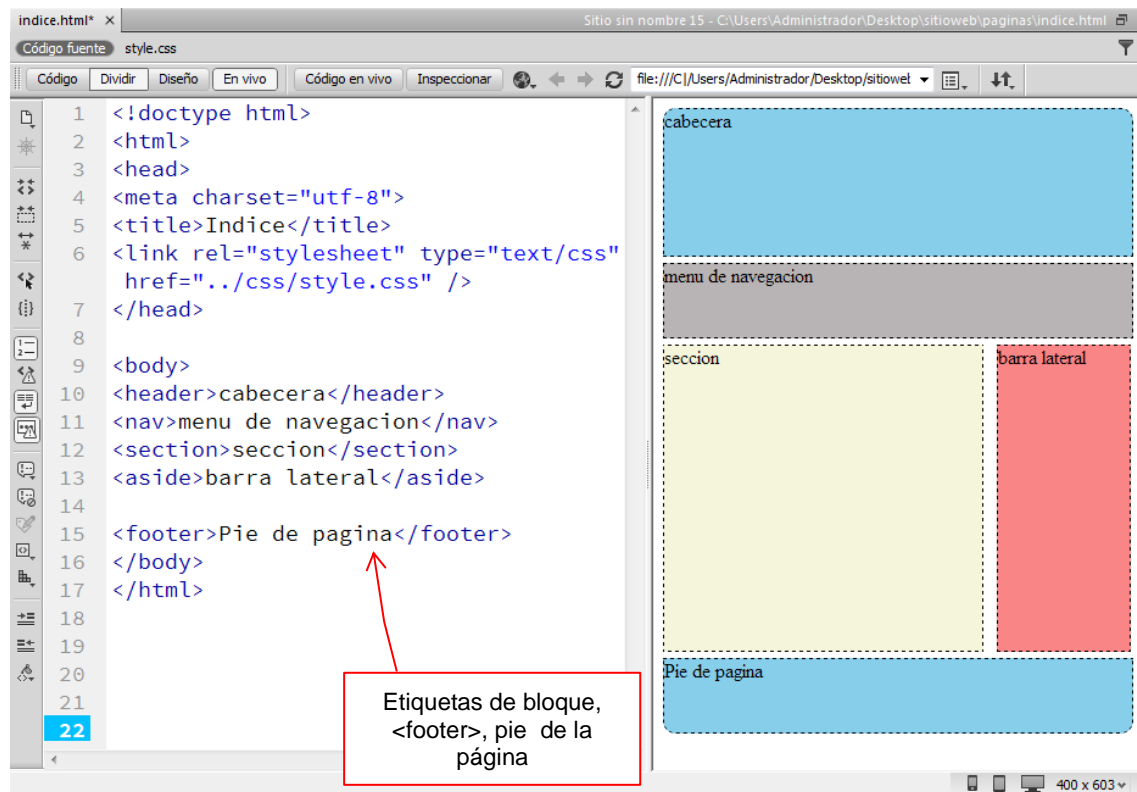
A continuación, guardar el archivo estilo.css: CTRL+S



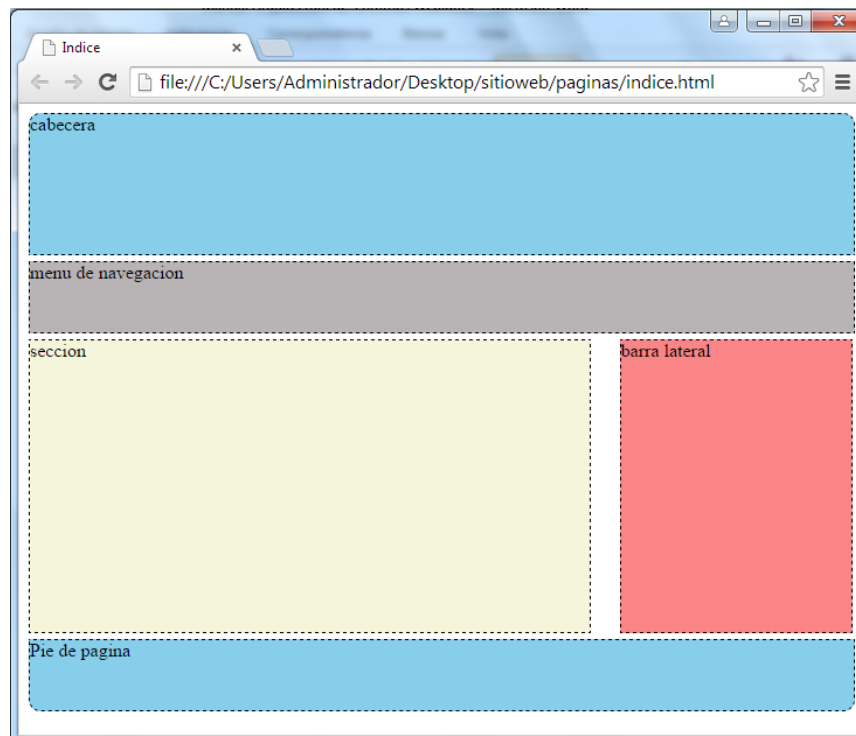


## Agregando las etiquetas de bloque en la página

Definida las reglas en style.css, dentro del body defina la etiqueta <footer>, tal como se muestra. Presiona la tecla F5 para visualizar el diseño de la página.



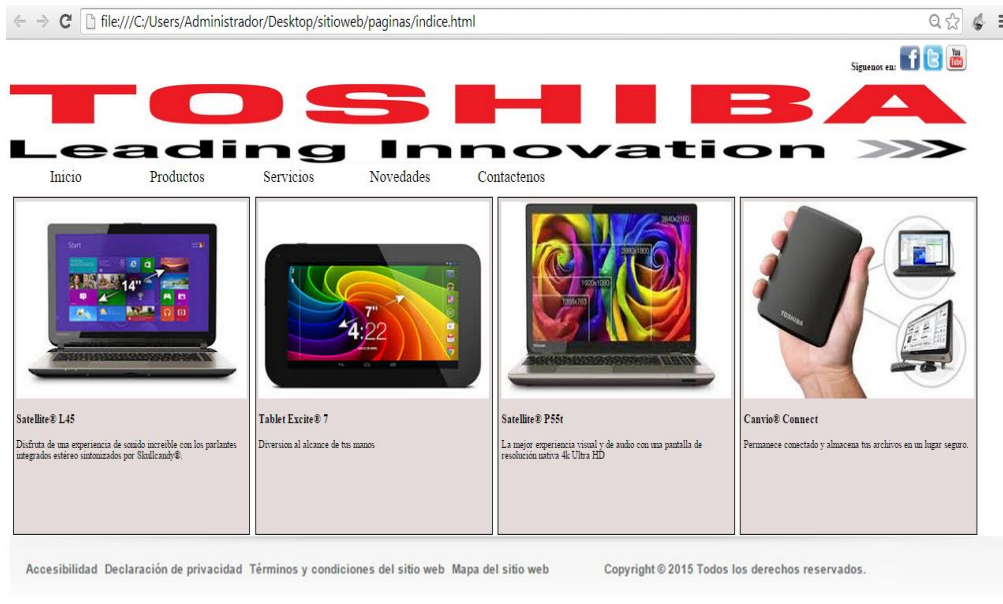
Presiona la tecla F12 para ejecutar la página por el navegador, tal como se muestra.



## LABORATORIO 5.2

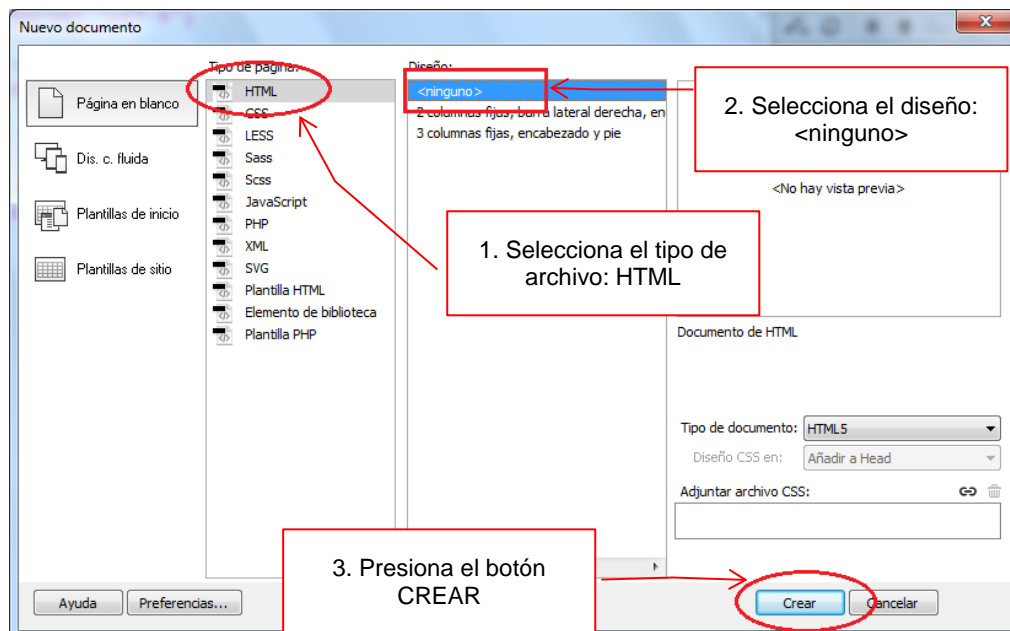
### Se nos pide diseñar una página de inicio de la empresa. Uo utilizando bloques y estilos CSS

Diseña una página de inicio HTML donde utilices las etiquetas <header>, <nav>, <section>, <aside>, <footer> y estilos CSS, tal como se muestra en la gráfica.

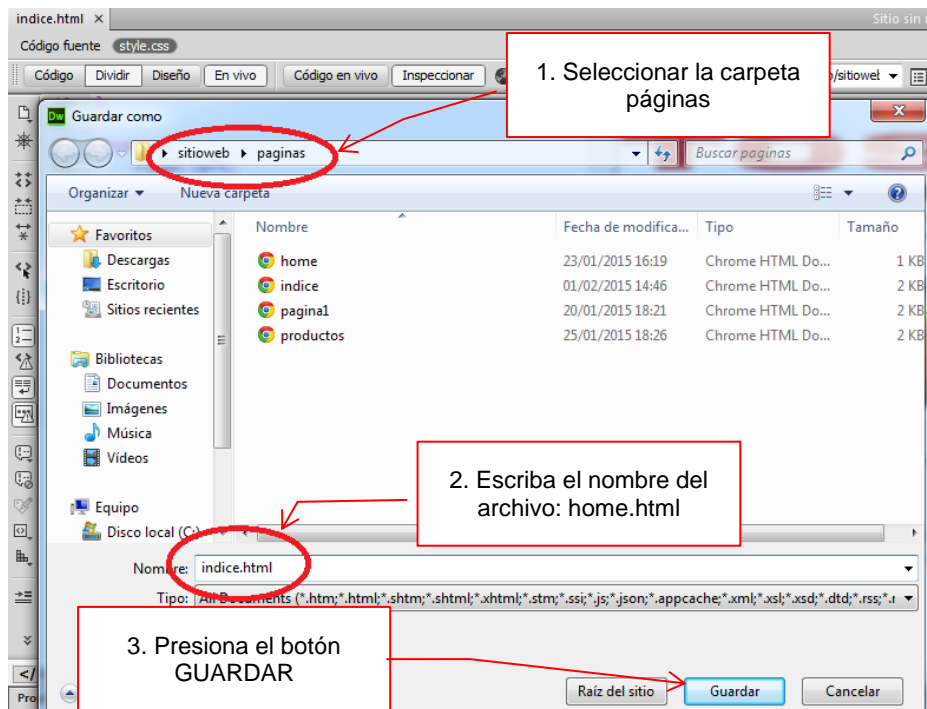


### Agregando la pagina HTML

A continuación, creamos un archivo HTML, el cual no tendrá diseño (ninguno), tal como se muestra.

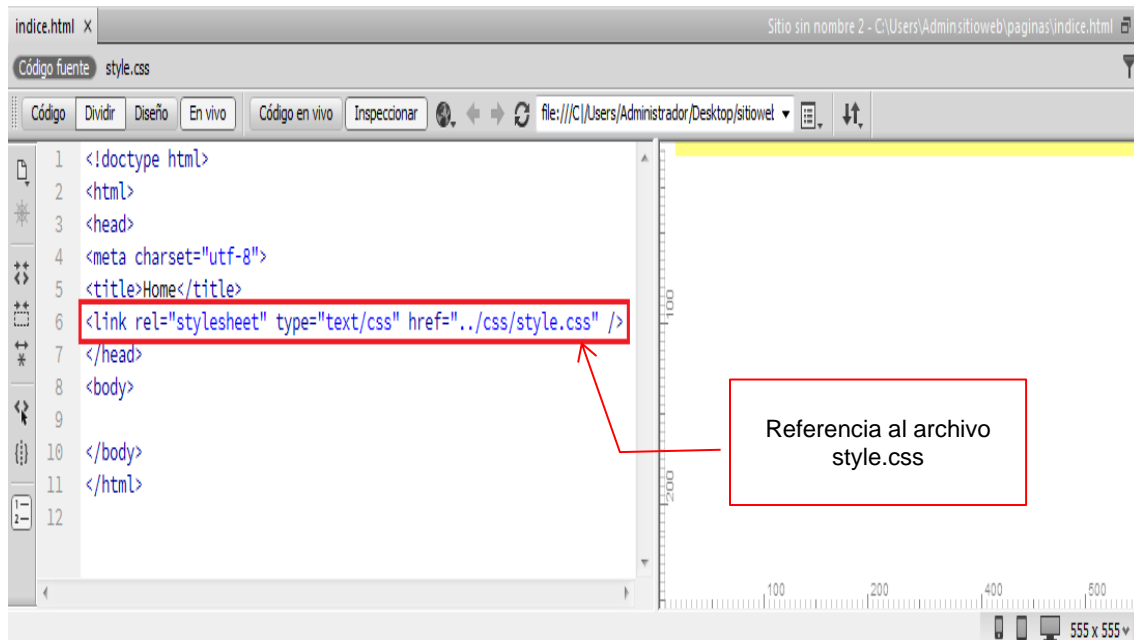


Creado el documento HTML, procedemos a guardarla, presiona la combinación CTRL+S



### Enlazando a la hoja de estilo

Creada la página, definimos, dentro del <head> la etiqueta <link> el cual hace una referencia al archivo CSS style.css, tal como se muestra.

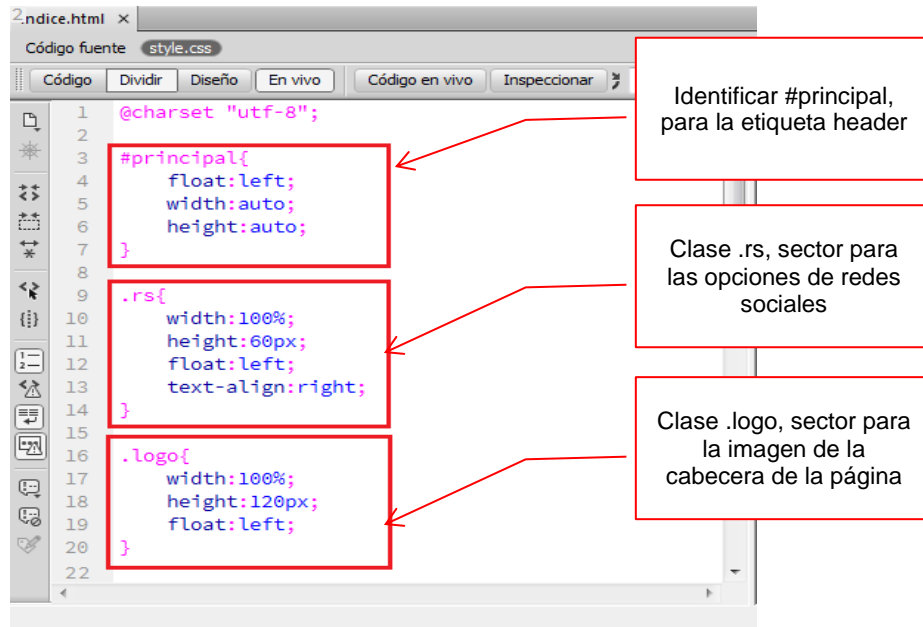


## Agregando reglas de estilo a la cabecera de la página

En el archivo CSS, defines las reglas de estilo para la cabecera:

- #principal, identificador que define las reglas para la etiqueta <header>
- .rs, clase que define el sector para las opciones de redes sociales
- .logo, clase que define el sector para la imagen de la página

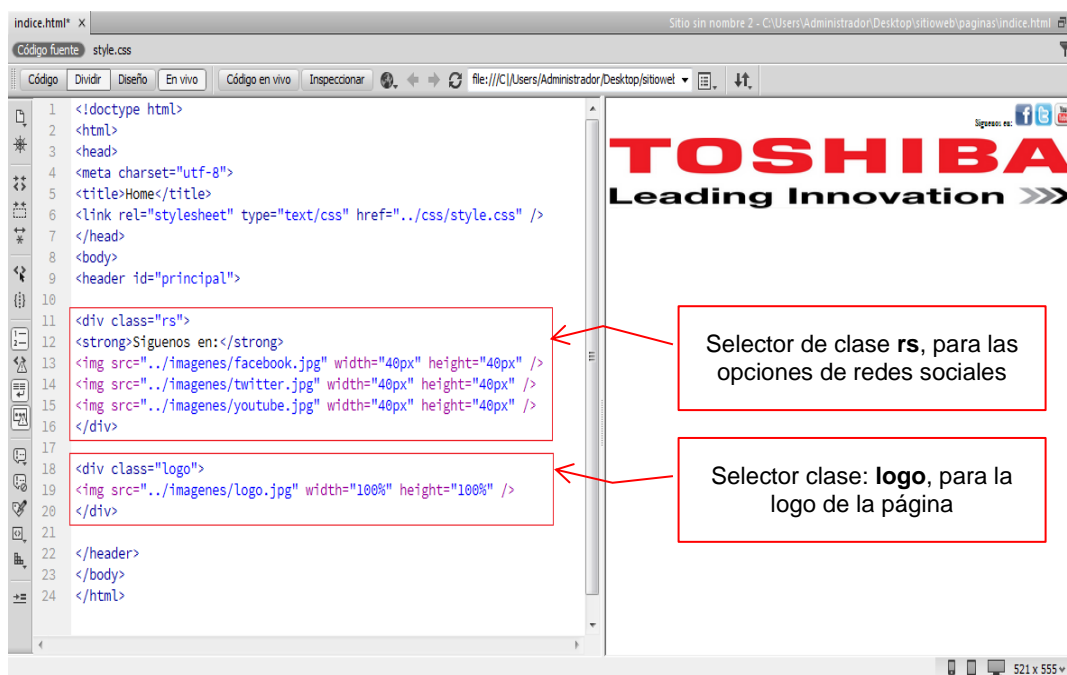
A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <header> y su contenido en la página HTML

Definida las reglas en style.css, dentro del body define la etiqueta <header> asignando el identificador **principal**. Agrega dos bloques div:

- <div class="rs"> sector para las opciones de redes sociales
- <div class="logo"> sector para el logo de la página.

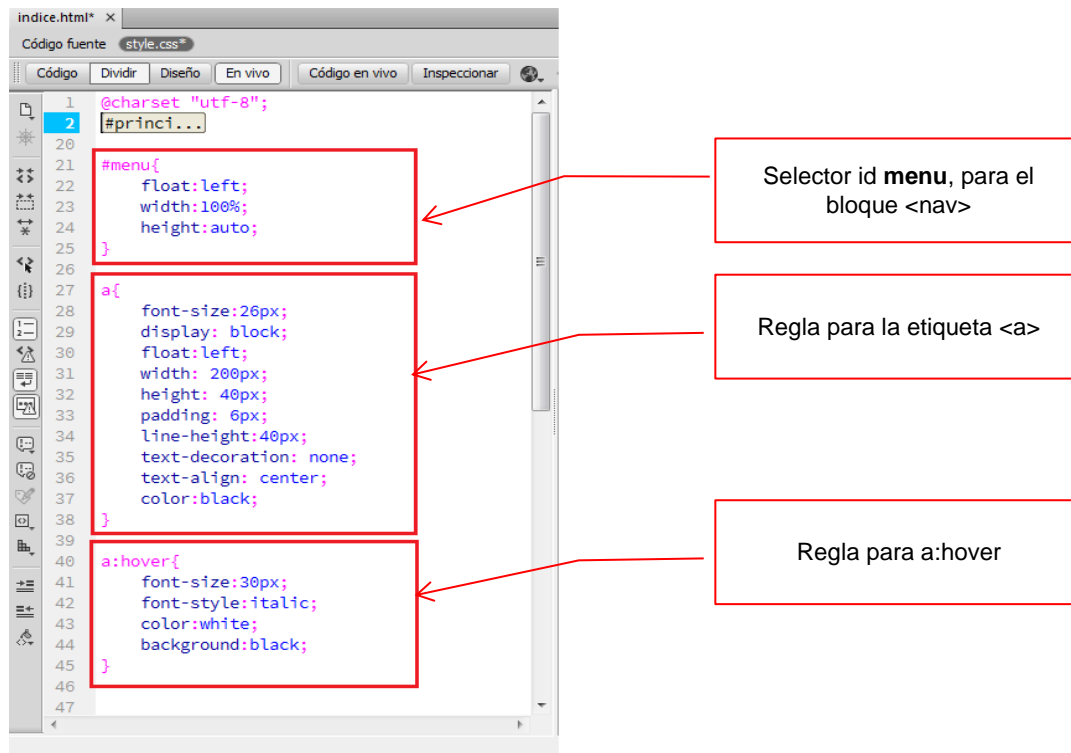


## Agregando las reglas en el Menu de Navegación

En el archivo CSS, defina las reglas de estilo para el menú de navegación:

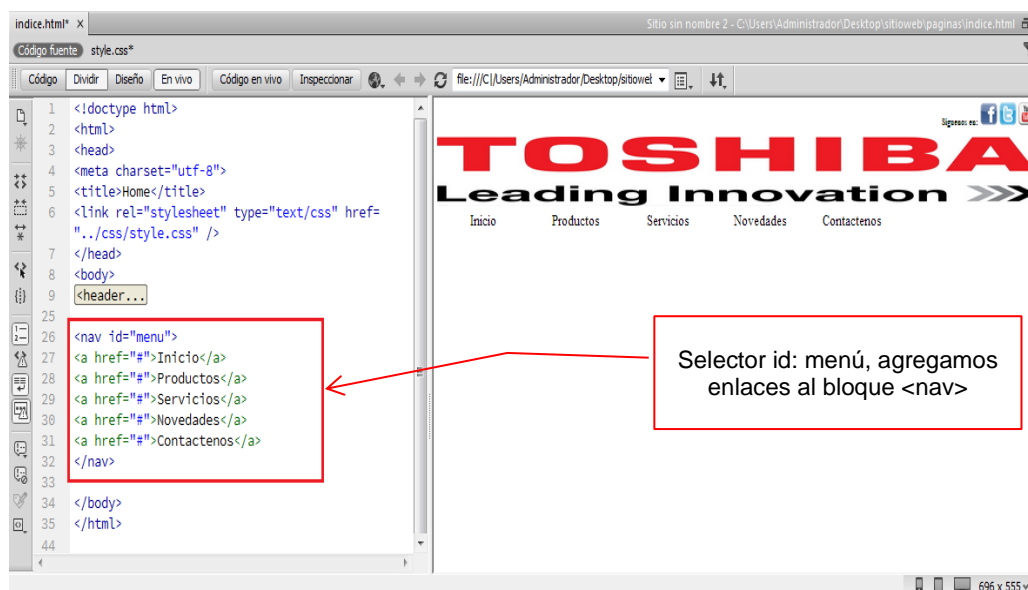
- #menu, identificador que define las reglas para la etiqueta <nav>
- a, regla para los enlaces en el menú de navegación
- a: hover, regla para el evento hover de la etiqueta <a>

A continuación guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <nav> y sus enlaces en la página HTML

Definida las reglas en style.css, defina la etiqueta <nav> agregando enlaces <a href>, tal como se muestra. Presiona la tecla F5 para visualizar el diseño de la página

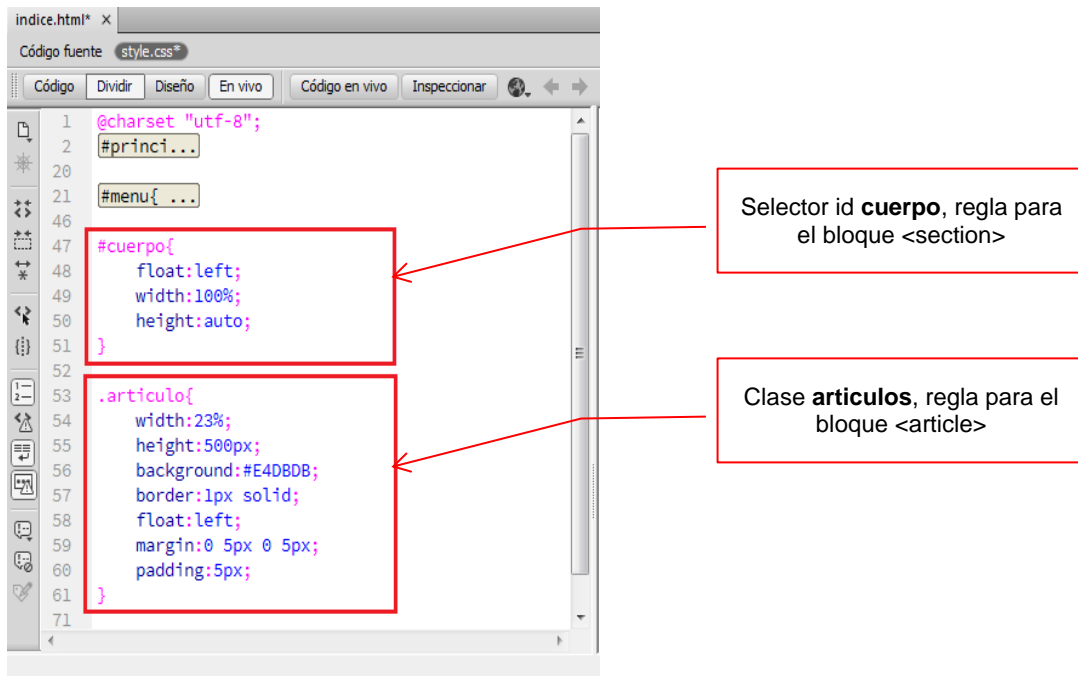


## Agregando las reglas en el cuerpo de la pagina <section>

En el archivo CSS, style.css defina las reglas para el cuerpo de la página

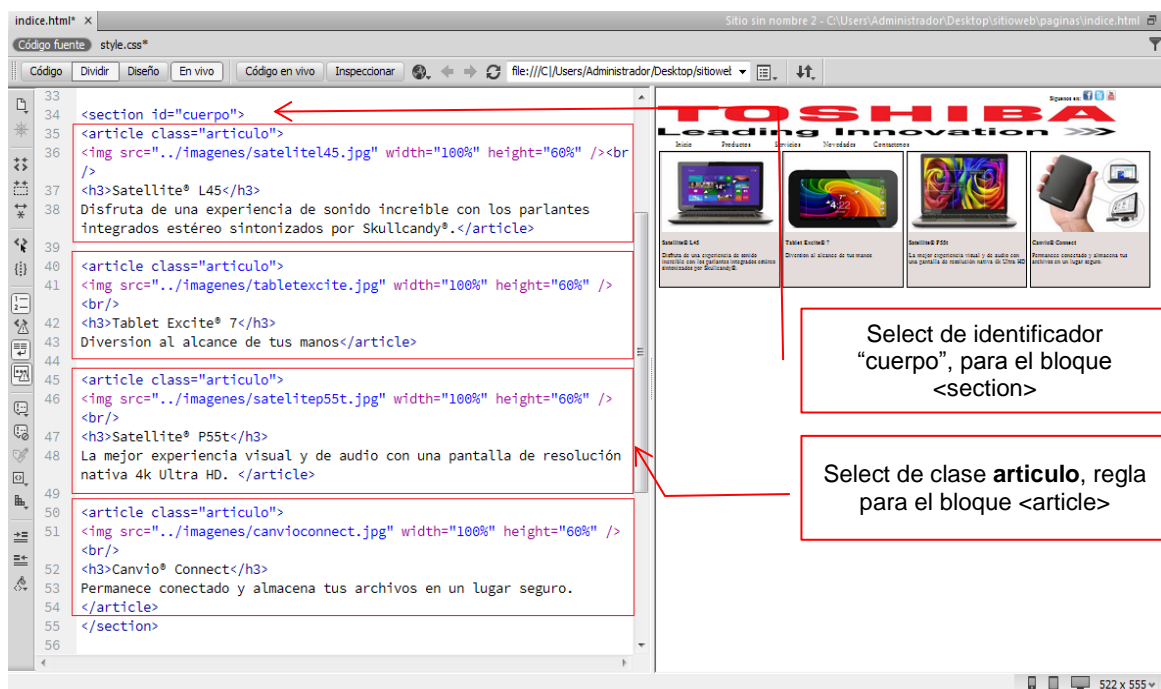
- #cuerpo, regla para el bloque <section>
- .articulo, regla para el bloque <article>

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <section> y <articulo> para la página HTML

Definida las reglas en style.css, defina la etiqueta <section> para promocionar los productos. Agregue bloques <article> definiendo su id="articulo". Presiona la tecla F5 para visualizar el diseño.

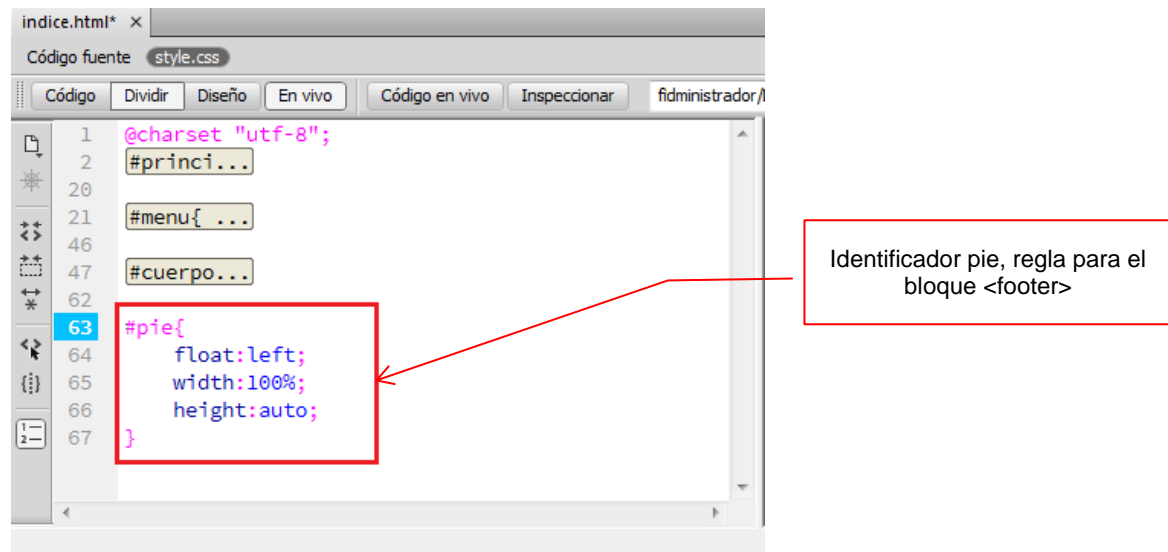


## Agregando las reglas al pie de página

En el archivo CSS, style.css definimos las reglas para el pie de página.

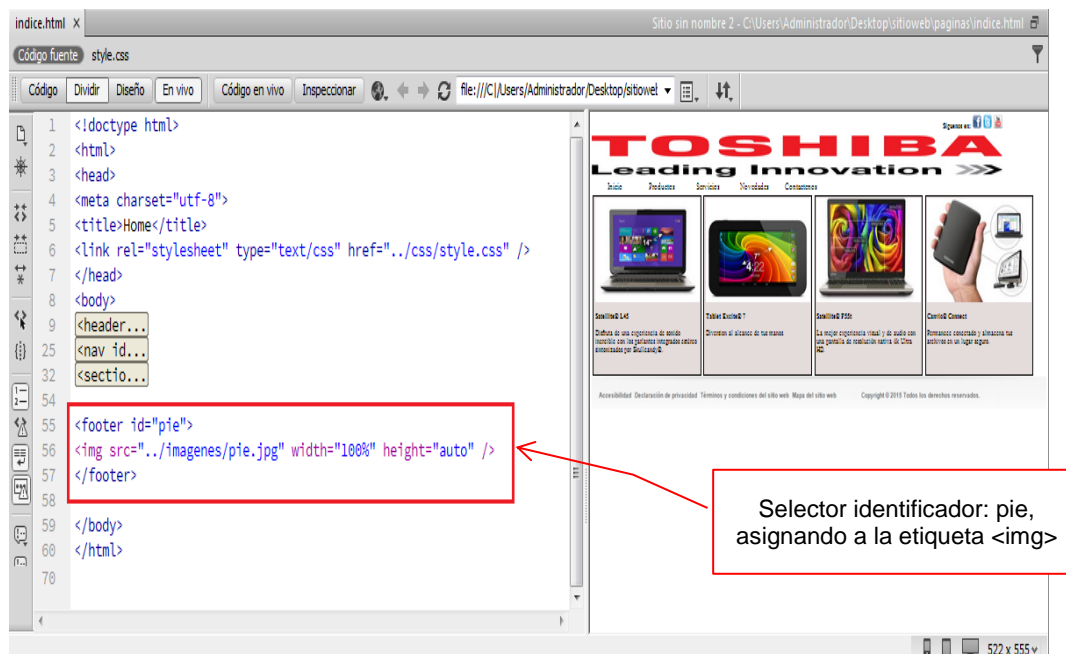
- **#pie**, identificador para el bloque <footer>

Acontinuación guardar el archivo estilo.css: CTRL+ S



## Agregando el pie de página <footer>

Definida las reglas en style.css, defina la etiqueta <footer> agregando su identificador **pie**, tal como se muestra. Presiona la tecla F5 para visualizar el diseño.



Para ejecutar a través del navegador, primero guardar la página: CTRL+S y presiona la tecla F12 para ejecutar la página con el navegador.



## LABORATORIO 5.3

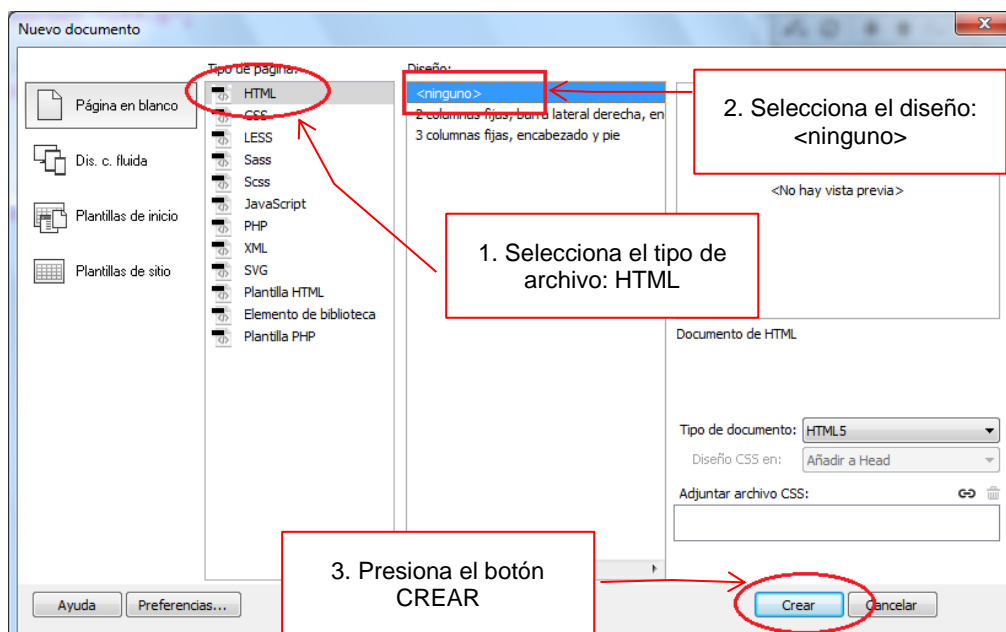
Se nos pide diseñar una página de promoción de productos (laptops) que vende la empresa, donde realizamos una breve descripción de algunos de los productos que ofrece la empresa, tal como se muestra. Utilice bloques y estilos CSS.



### Agregando la pagina HTML

A continuación, creamos un archivo HTML, el cual no tendrá diseño (ninguno), tal como se muestra.

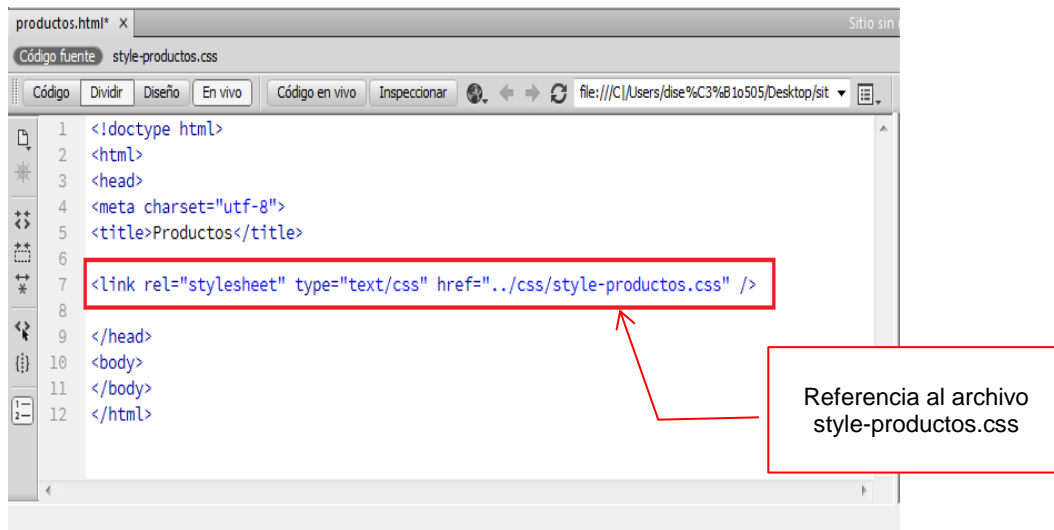
Guardar la pagina con el nombre de productos.html, en la carpeta paginas.





## Enlazando a la hoja de estilo

Creada la página, definimos, dentro del <head> la etiqueta <link> el cual hace una referencia al archivo CSS style-producto.css, tal como se muestra.

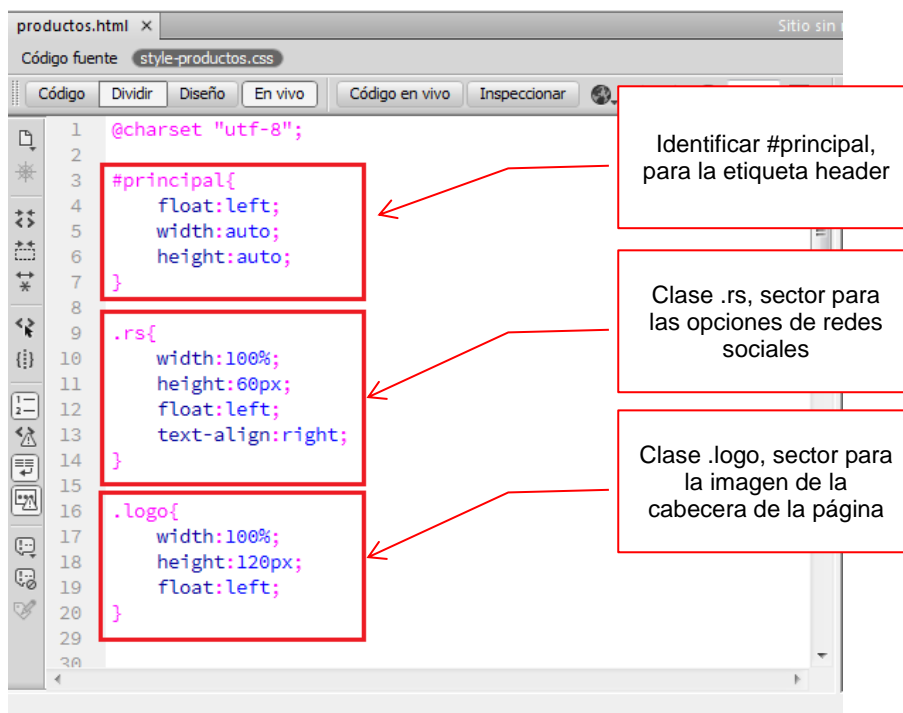


## Agregando reglas de estilo a la cabecera de la página

En el archivo CSS, defines las reglas de estilo para la cabecera:

- #principal, identificador que define las reglas para la etiqueta <header>
- .rs, clase que define el sector para las opciones de redes sociales
- .logo, clase que define el sector para la imagen de la página

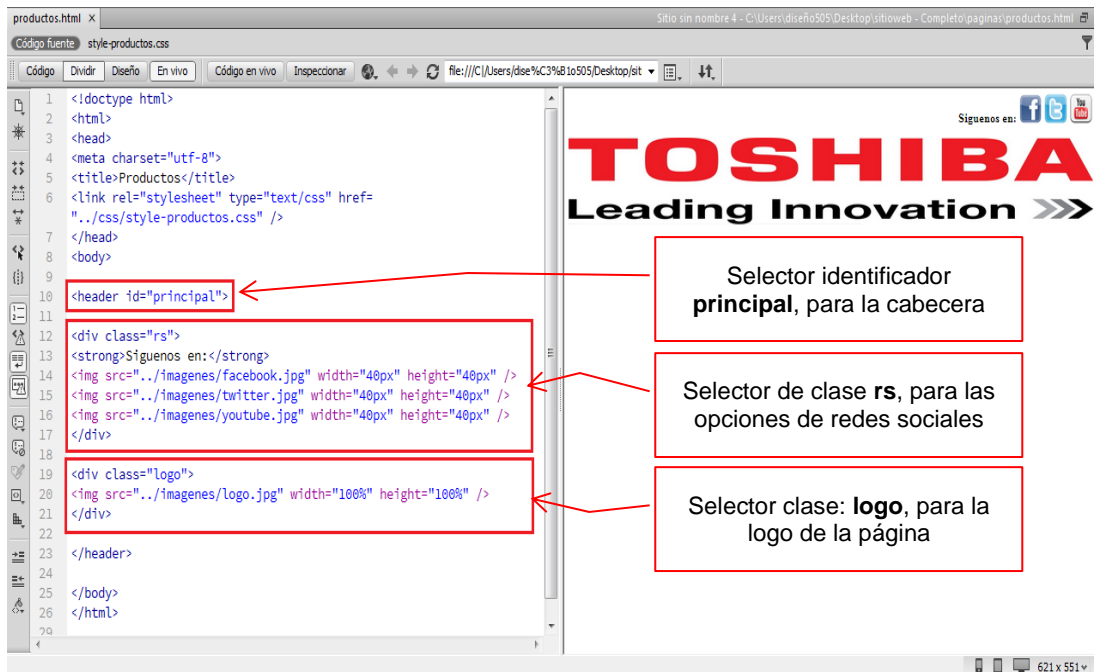
A continuación guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <header> y su contenido en la pagina HTML

Definida las reglas en style-productos.css, dentro del body defina la etiqueta <header> asignando el identificador **principal**. Agrega dos bloques div:

- <div class="rs"> sector para las opciones de redes sociales
- <div class="logo"> sector para el logo de la pagina.

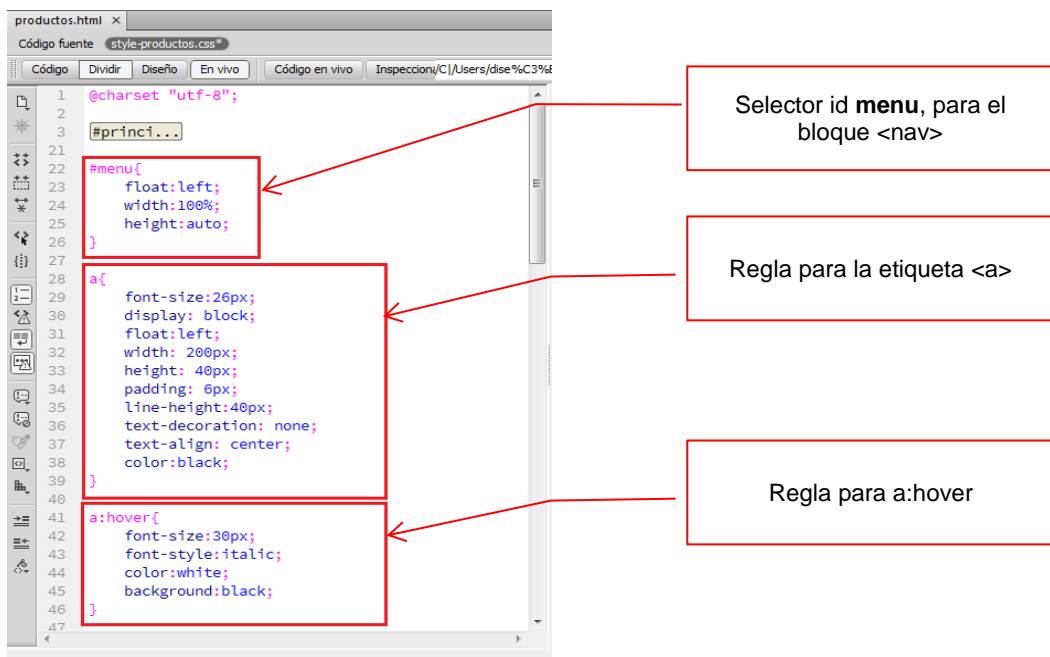


## Agregando las reglas en el Menu de Navegación

En el archivo CSS, defina las reglas de estilo para el menú de navegación:.

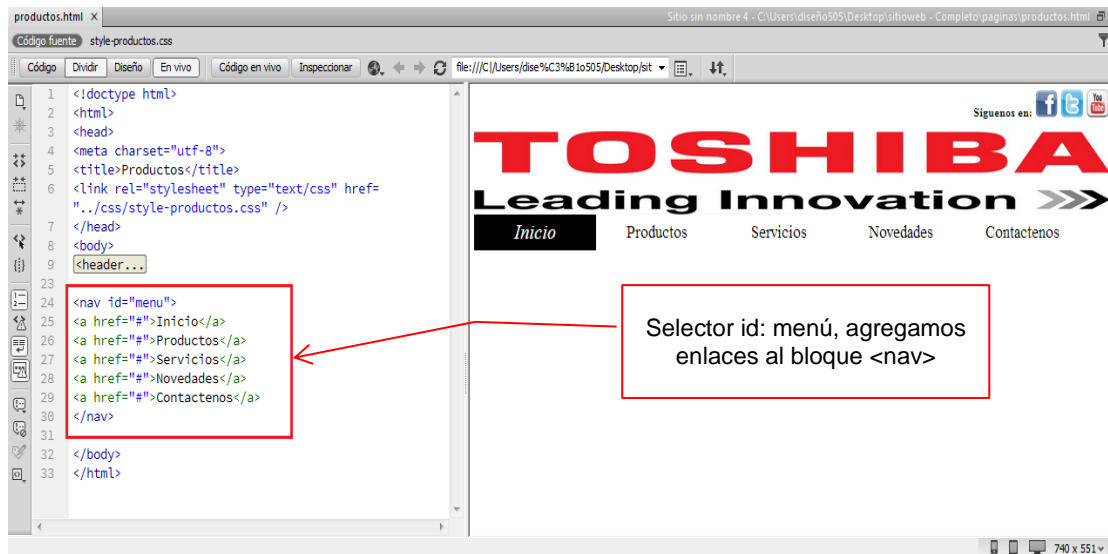
- #menu, identificador que define la reglas para la etiqueta <nav>
- a, regla para los enlaces en el menú de navegación
- a:hover, regla para el evento hover de la etiqueta <a>

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <nav> y sus enlaces en la página HTML

Definida las reglas en style-productos.css, defina la etiqueta <nav> agregando enlaces <a href>. Presiona la tecla F5 para visualizar el diseño de la página.

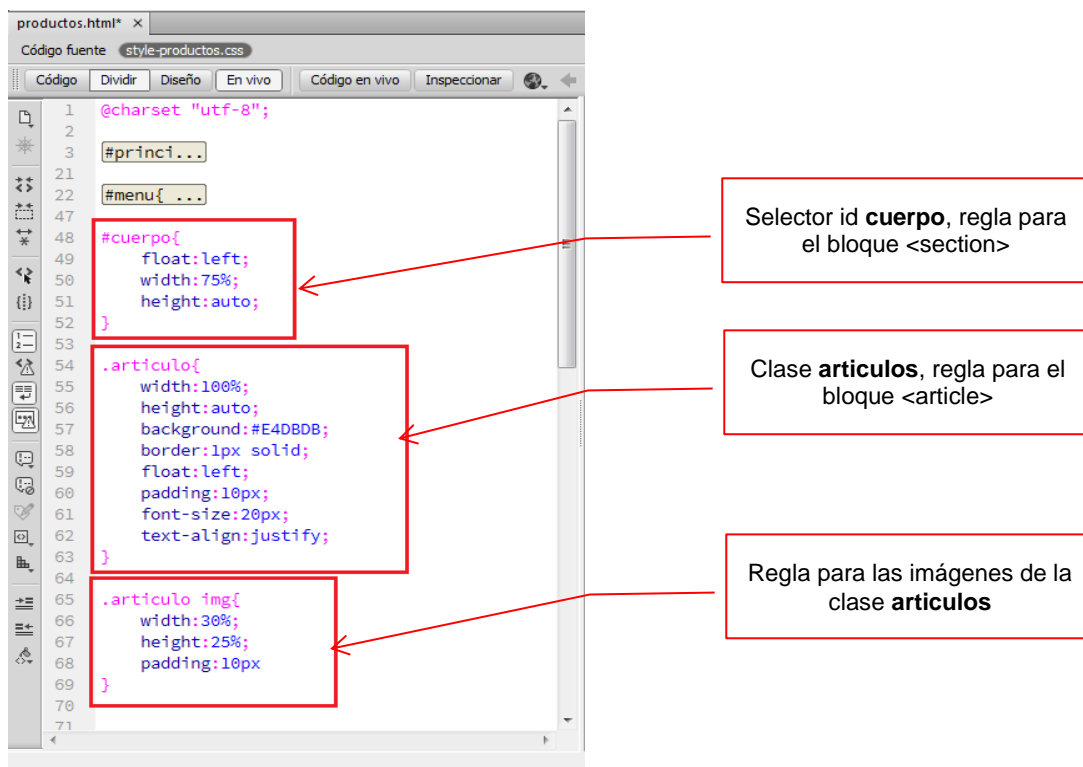


## Agregando las reglas en el cuerpo de la pagina <section>

En el archivo CSS, style-productos.css defina las reglas para el cuerpo de la página

- #cuerpo, regla para el bloque <section>
- .articulo, regla para el bloque <article>
- .articulo img, regla para las imágenes dentro de cada bloque article

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <section> y <articulo> para la página HTML

Definida las reglas en style-productos.css, defina la etiqueta <section> para promocionar los productos. Agregue bloques <article> definiendo su id="articulo". Presiona la tecla F5 para visualizar el diseño.

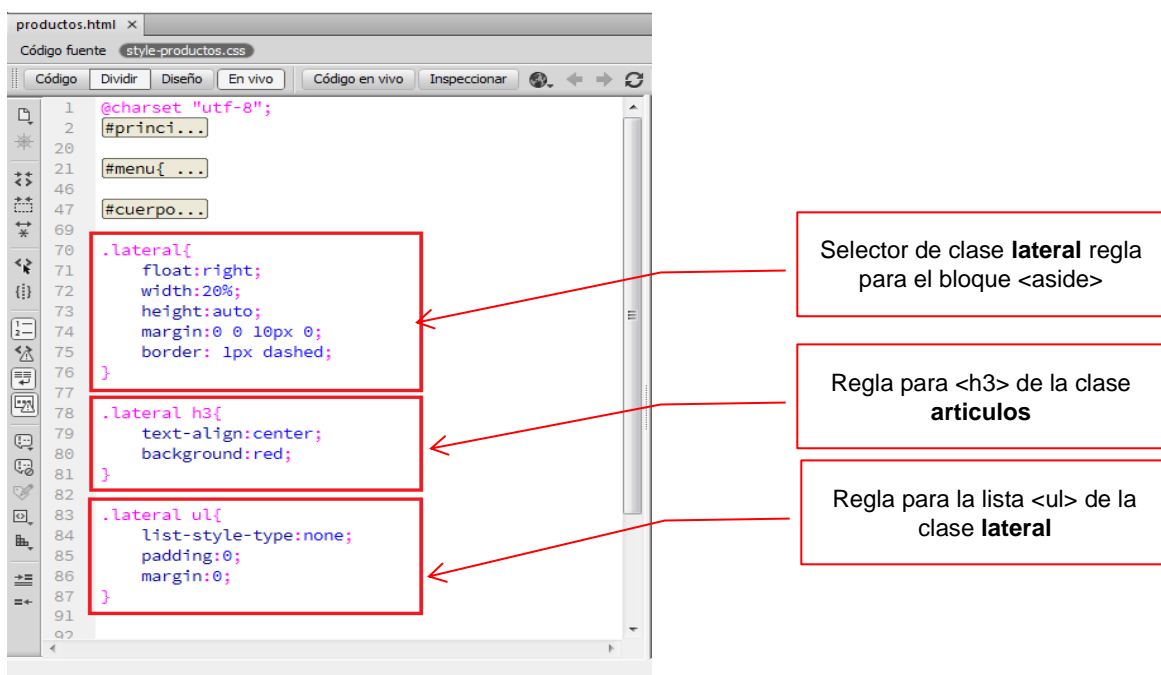


## Agregando las reglas en la barra lateral <aside>

En el archivo CSS, style-productos.css defina las reglas para barra lateral.

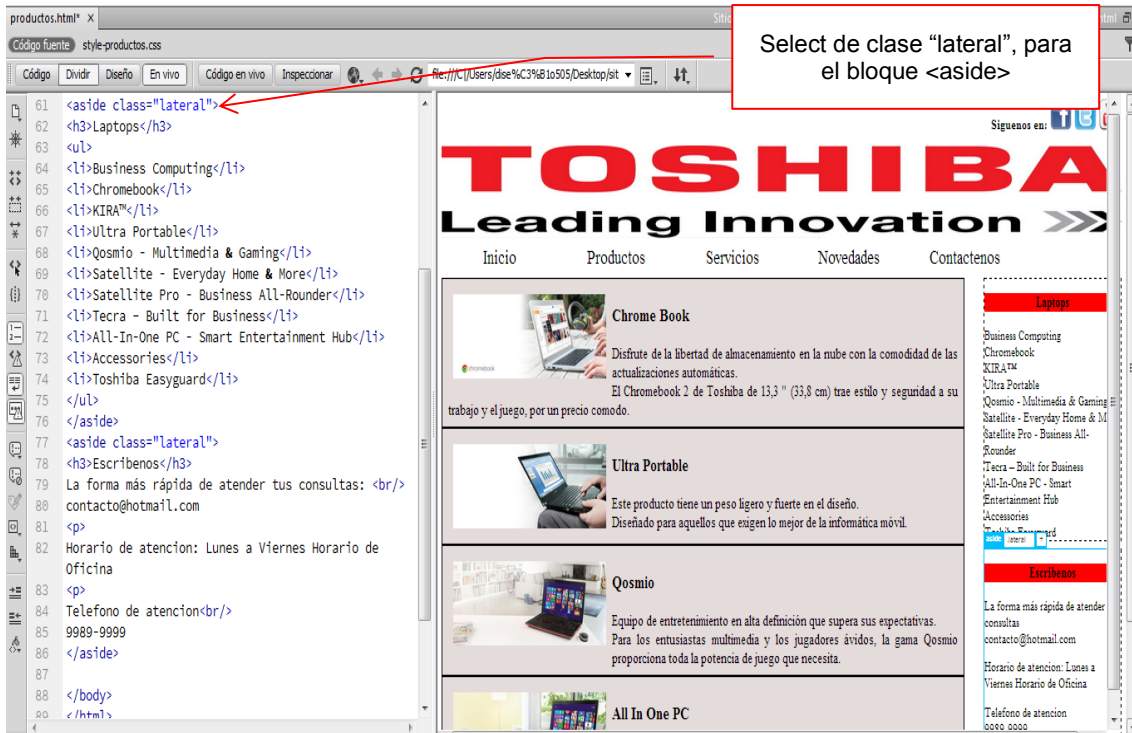
- .lateral, regla para el bloque <aside> donde su posición flotante es derecha
- .lateral h3, regla para la etiqueta <h3> del bloque <aside>
- .lateral ul, regla para la lista <ul> dentro de cada bloque article

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <aside> para la página HTML

Definida las reglas en style-productos.css, defina la etiqueta <aside> para mostrar información adicional de los productos y de contactos. Presiona la tecla F5 para visualizar el diseño.

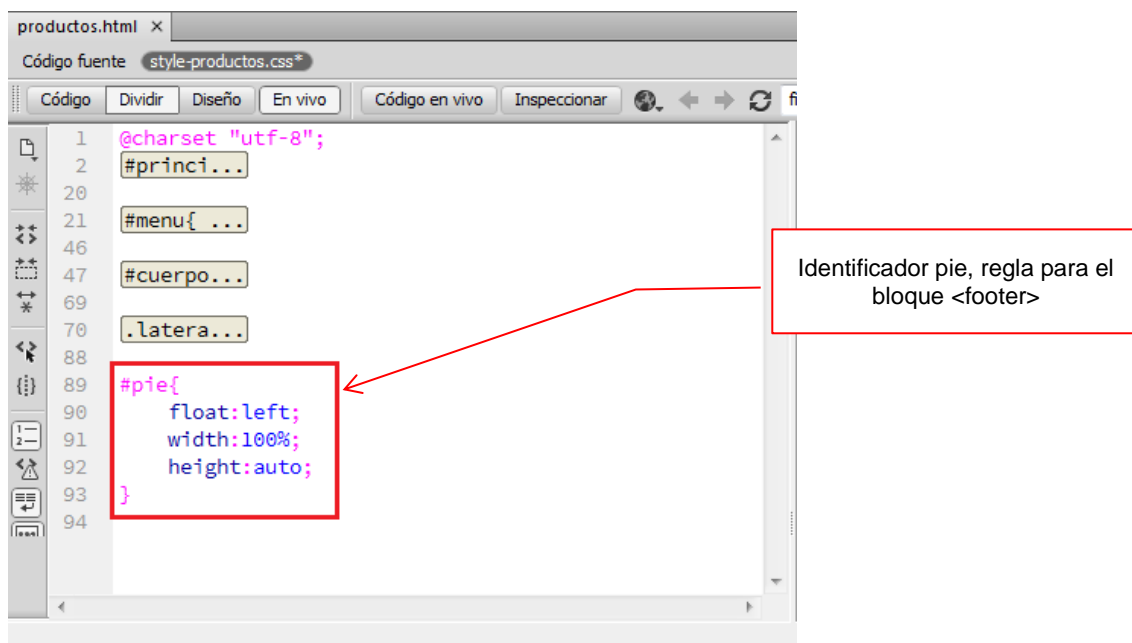


## Agregando las reglas al pie de página

En el archivo CSS, style.css definimos las reglas para el pie de página.

- #pie, identificador para el bloque <footer>

A continuación, guardar el archivo estilo.css: CTRL+ S



## Agregando el pie de página <footer>

Definida las reglas en style.css, defina la etiqueta <footer> agregando su identificador **pie**, tal como se muestra. Presiona la tecla F5 para visualizar el diseño.



Para ejecutar a través del navegador, primero guardar la página: CTRL+S y presiona la tecla F12 para ejecutar la página con el navegador.

# Resumen

- 📖 El modelo de cajas es el comportamiento de estilo CSS que hace que todos los elementos de una página se representen mediante cajas rectangulares. Este modelo es la característica más importante del lenguaje de estilos CSS, ya que condiciona el diseño de todas las páginas web.
- 📖 HTML5 nos aporta una serie de etiquetas nuevas que permiten mejorar la semántica de nuestra página. Los elementos de HTML5 nuevos: header, nav, section, article, aside, footer, hgroup, figure.
- 📖 Las partes que compone un bloque y su orden de visualización desde el punto de vista del usuario son: content se trata del contenido HTML, width, controla el ancho de la caja, height, controla la altura de la caja, border, línea que encierra completamente el contenido y su relleno, margin, separación opcional entre la caja y el resto de cajas adyacentes.
- 📖 Las nuevas propiedades CSS3 son extremadamente poderosas y deben ser estudiadas una por una, pero para facilitar su aprendizaje vamos a conocerlas: padding, espacio libre opcional existente entre el contenido y el borde, background-image, imagen que se muestra por detrás del contenido y el espacio de relleno; background-color, color que se muestra por detrás del contenido y el espacio de relleno; border-radius, efecto de las esquinas redondeadas; float, establece el posicionamiento flotante de la caja; clear, modifica el comportamiento de las cajas flotantes.
- 📖 Para aplicarla efectivamente estas propiedades a sus documentos, debe declararla con los correspondientes prefijos para los navegadores más comunes son los siguientes: -moz- para Firefox; -webkit- para Safari y Chrome; -o- para Opera; -khtml- para Konqueror; -ms- para Internet Explorer; -chrome- específico para Google Chrome
- 📖 Las propiedades **display** y **visibility** controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página. Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.
- 📖 La propiedad **box-sizing** nos permite cambiar cómo el espacio total ocupado por un elemento en pantalla será calculado forzando a los navegadores a incluir en el ancho original los valores de las propiedades padding y border. La propiedad **box-sizing** puede tomar dos valores. Por defecto es configurada como **content-box**, lo que significa que los navegadores agregarán los valores de padding y border al tamaño especificado por **width** y **height**.
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 [http://librosweb.es/css/capitulo\\_4.html](http://librosweb.es/css/capitulo_4.html)

🔗 <http://www.desarrolloweb.com/articulos/aprende-html5-5-minutos.html>

🔗 <http://www.tutosytips.com/hagamos-un-sitio-en-html5-y-css3-parte-1/>

🔗 <http://blog.ikhuerta.com/css3-displaybox-maquetacion-con-el-modelo-de-caja-flexible>







## ESTRUCTURAS Y LAYOUT

---

### LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3 y estructuras.

### TEMARIO

#### 3.1 Tema 5: Creacion de Bloques

- 3.1.1 Etiquetas de bloques: header, nav, aside, section, article, footer, div.
- 3.1.1.2 Estilos básicos del bloque: width, height, background, border, margin.
- 3.1.1.2 Estilos para bloques: padding, background-image, border-radius, float, clear
- 3.1.1.3 Propiedad box-sizing

#### 3.2 Tema 6: Creacion de Menú

##### 3.2.1 Estilos a vínculos: crecion de menú

##### 3.2.1 Ejercicio Integrador

### ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5, hojas de estilo CSS3 y bloques de contenidos
- Los alumnos diseñan un menú en una página web



## 3.2 CREACION DE MENU

HTML5 y CSS3 son dos recursos esenciales para diseñadores y desarrolladores en la actualidad si lo que se busca es crear sitios web funcionales, así como estéticamente hermosos. Estos lenguajes permiten crear páginas web con efectos realmente alucinantes e innovadores y además son muy simples de usar.



Figura 1

Referencia: <http://www.cssblog.es/menu-con-varios-niveles-mediante-css/>

Otro elemento bastante típico y común en muchas páginas web es la zona de navegación, formada por enlaces a las zonas principales de la web, o por un menú de navegación, desplegable de enlaces, etc. Por lo visto Html5 interpreta que es una zona que deberíamos identificar y ha creado para ello la nueva etiqueta `<nav>` `</nav>`. En internet explorer, por desgracia este elemento `<nav>` no es compatible, sin embargo, hay una solución sencilla que se puede utilizar para corregir el problema y que el menú se despliegue sin errores de visualización.

No todos los enlaces de una página web han de pertenecer obligatoriamente a un nav. Los enlaces de políticas, de datos de contacto, mapas del sitio, copyrights, enlaces a secciones secundarias etc, que aparecen normalmente en un pie de página no es necesario que los rodeemos con la etiqueta `<nav>`, pues la etiqueta `<footer>` ya recoge esos aspectos. Ésta etiqueta es más bien para enlaces que favorezcan la navegación por la web, que nos dirijan a zonas importantes.

El por qué HTML5 añade esta etiqueta es sencillo. Cada vez es más habitual que nuestras páginas aparezcan no solo en los navegadores que usamos en el ordenador, sino también en dispositivos móviles, teléfonos, iPads, etc. Al usar esta etiqueta el agente de usuario (llamado también navegador) podría dar la opción al usuario de esconder el menú de navegación para no molestar, para tener más espacio disponible para ver el contenido importante. También podría dar la opción de hacerlo aparecer cuando el usuario quiera cambiar de sección. Esto solo sería posible si se sabe dónde empieza y dónde termina ese menú de navegación y eso es precisamente lo que conseguimos con la etiqueta `<nav>`.

En la norma donde se dictan todas estas reglas del HTML5 no se indica que no pueda existir más de un bloque del tipo `<nav>` dentro de una misma página. No obstante, en algunos sitios que se recomienda que solo exista uno.

### 3.2.1 Estilo a vínculos: creación de menú

Para diseñar un menú de navegación hacemos uso de listas, porque un menú básicamente es una lista, no es un párrafo, ni un título, es una lista de vínculos.

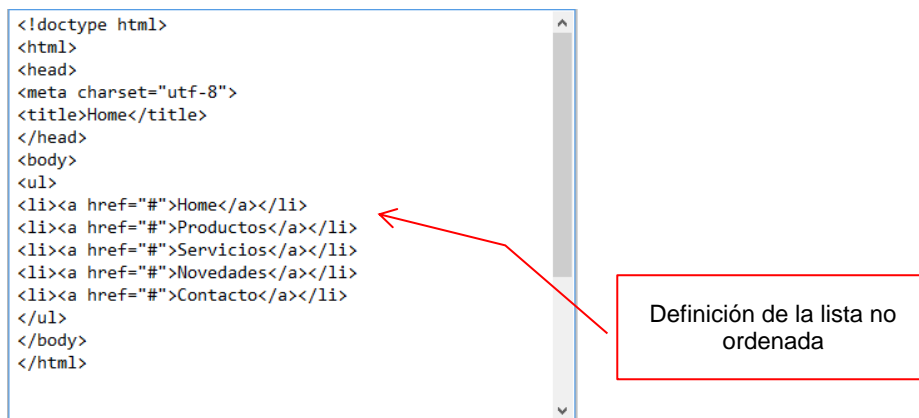
Para cambiar el aspecto de las listas, CSS nos permite realizar cambiar el aspecto de la lista, para convertirla en el bonito menú que queremos, sin que deje de ser lista. Pero primero, veamos en qué consisten los pre-formatos que trae de por sí una lista.

Las listas se renderizan en el navegador con ciertas características, como el margen/relleno (los cuales son interpretados de diferente manera en los distintos navegadores como opera, Chrome, Internet, entre otros) de la lista completa, el margen / relleno (idem) de cada item de la lista y las viñetas, además de un salto de línea para cada item por ser elementos de bloque.

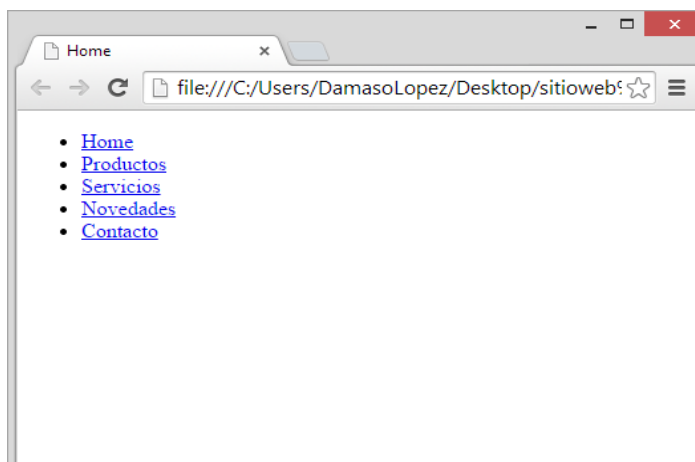
#### Creando un menú de navegación

A continuación, vamos a crear un menú con HTML realizando algunas modificaciones utilizando CSS

Primero creamos la lista no ordenada en la página HTML, tal como se muestra.



Guardar el archivo como página HTML. Abrir el documento con el navegador Chrome, donde se visualiza la página.



A continuación le quitamos las viñetas y márgenes de la lista en la etiqueta `<ul>` a través de la propiedad `list-style:none`

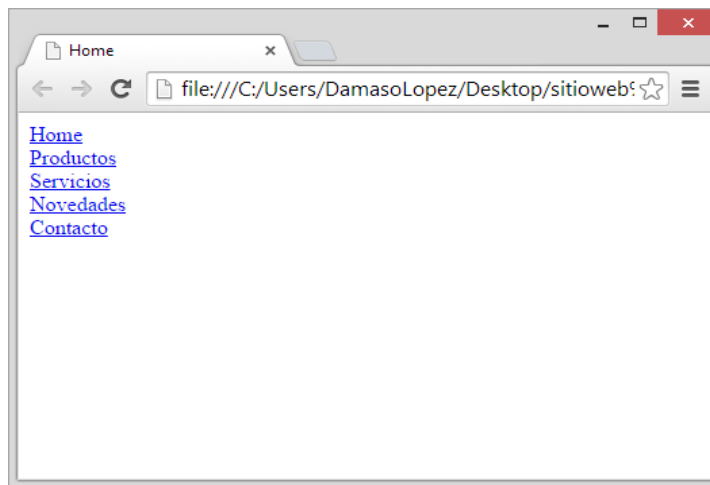
```
<style>

ul{
    list-style-type:none;
    margin:0;
    padding:0;
}

</style>
```

Quitamos las viñetas a la lista, no márgenes

Actualizar el documento en el navegador Chrome, visualizando la página.



Para que el menú sea horizontal, en la etiqueta `<li>` de la lista, agregamos la propiedad `float:left`, para hacer que los elementos de la lista se coloquen uno al lado del otro.

```
<style>

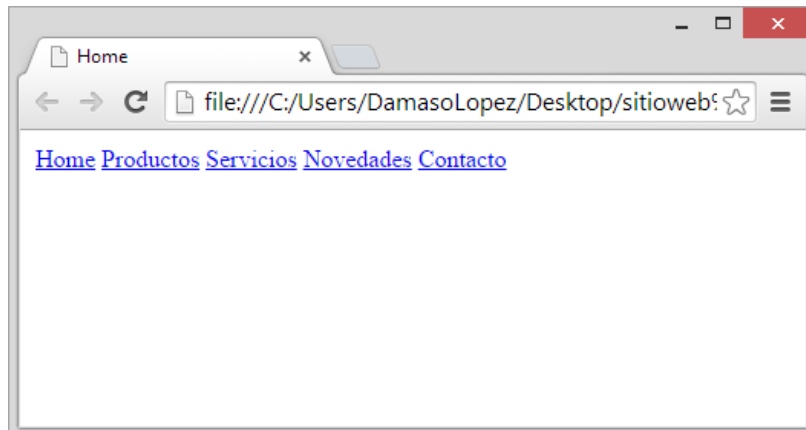
ul{
    list-style-type:none;margin:0;padding:0;
}

ul li{
    margin:2px;
    padding:0;
    float:left;
}

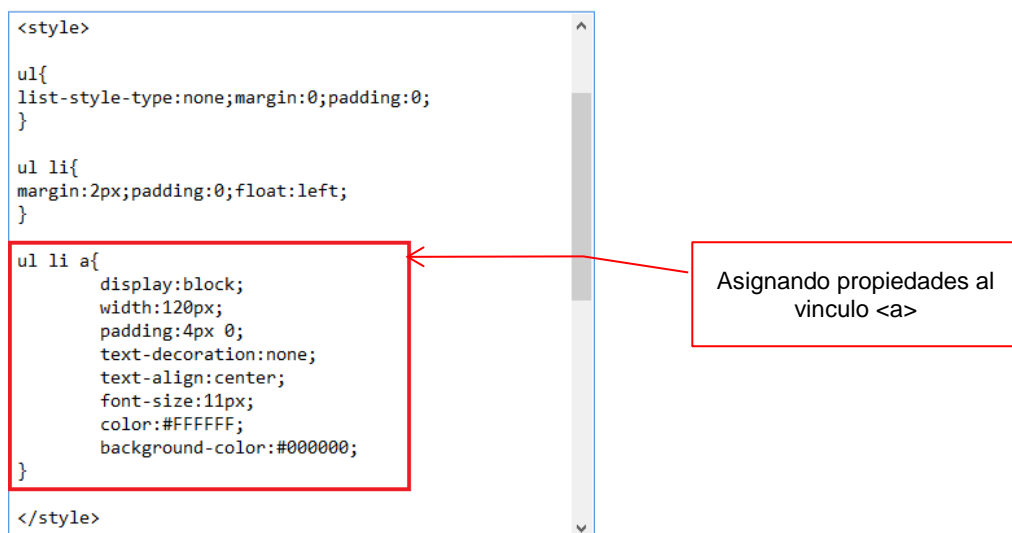
</style>
```

Uso de la propiedad `float:left`

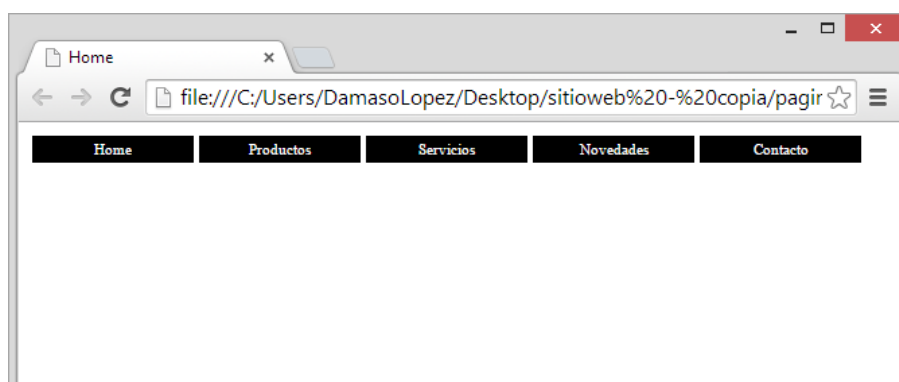
Actualizar el documento en el navegador Chrome, visualizando la página.



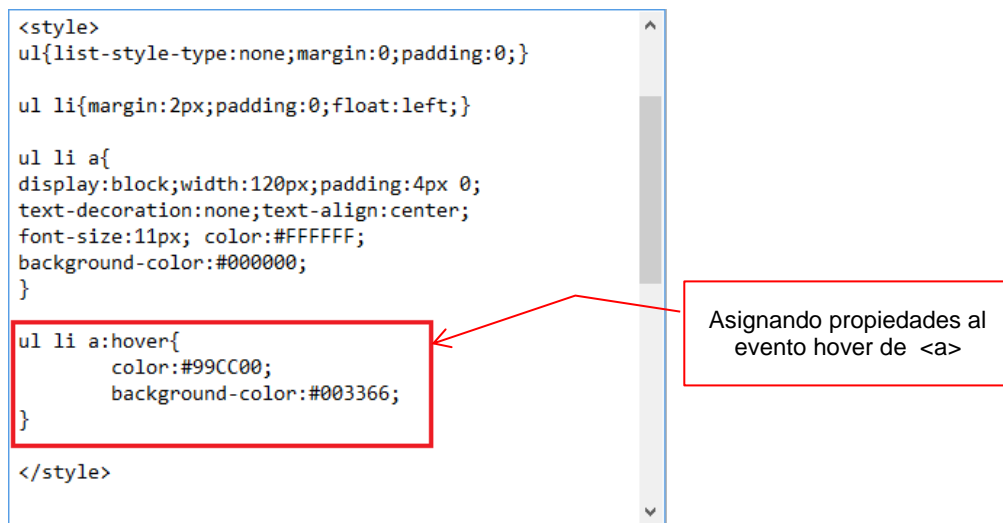
A continuación definimos estilos directamente sobre los vínculos, <a> asignando: bloque, tamaño, tipo de letra, colores, decoración, etc. Para seleccionar los vínculos dentro de los elementos de lista, encadenamos los nombres de las etiquetas en el selector CSS como lo hicimos con los <li>:



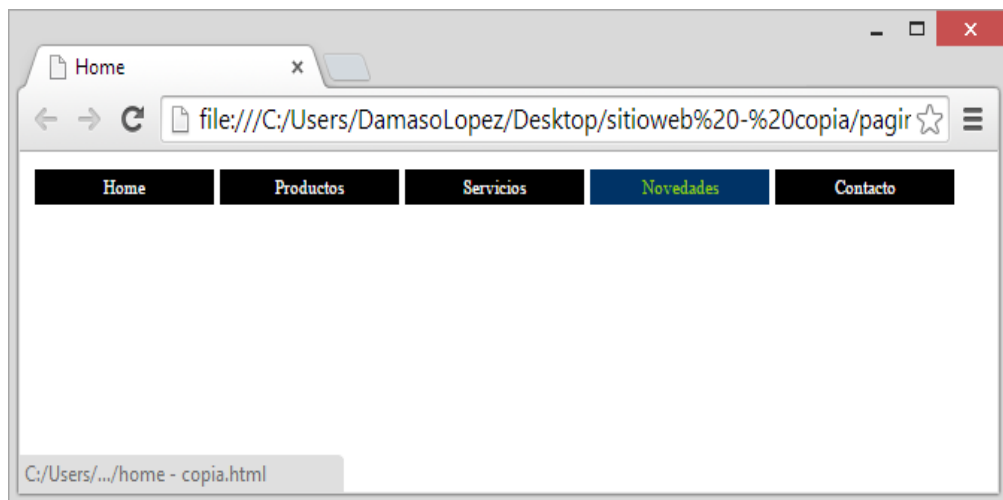
Actualizar el documento en el navegador Chrome, visualizando la página.



Por último, definimos propiedades del evento hover de la etiqueta <a>, para cambiar el color de letra y de fondo al acercar el cursor en el vínculo.



Actualizar el documento en el navegador Chrome, visualizando la página.



### **Creando un menú desplegable de navegación**

A continuación vamos a crear un menú desplegable con HTML realizando algunas modificaciones utilizando CSS.

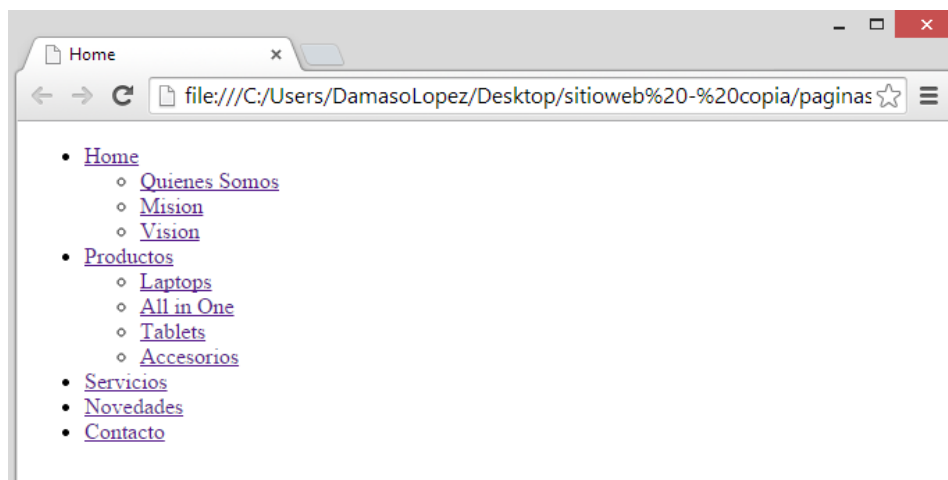
Primero creamos la lista no ordenada en la página HTML, tal como se muestra

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Home</title>
</head>
<body>

<ul class="menu">
<li><a href="#">Home</a>
  <ul>
    <li><a href="#">Quienes Somos</a></li>
    <li><a href="#">Mision</a></li>
    <li><a href="#">Vision</a></li>
  </ul>
</li>
<li><a href="#">Productos</a>
  <ul>
    <li><a href="#">Laptops</a></li>
    <li><a href="#">All in One</a></li>
    <li><a href="#">Tablets</a></li>
    <li><a href="#">Accesorios</a></li>
  </ul>
</li>
<li><a href="#">Servicios</a></li>
<li><a href="#">Novedades</a></li>
<li><a href="#">Contacto</a></li>
</ul>

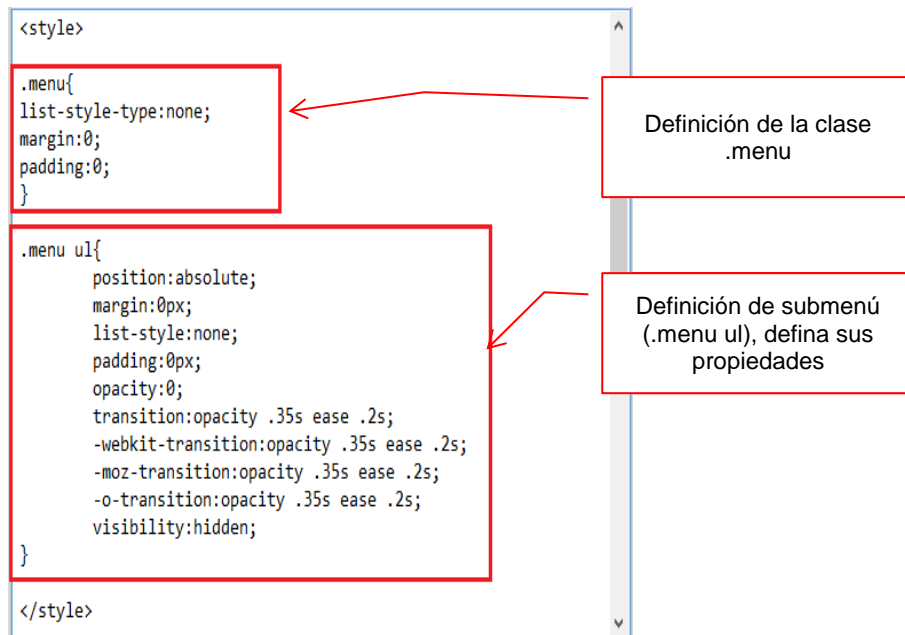
</body>
</html>
```

Guardar el archivo como pagina HTML. Abrir el documento con el navegador Chrome, donde se visualiza la página.

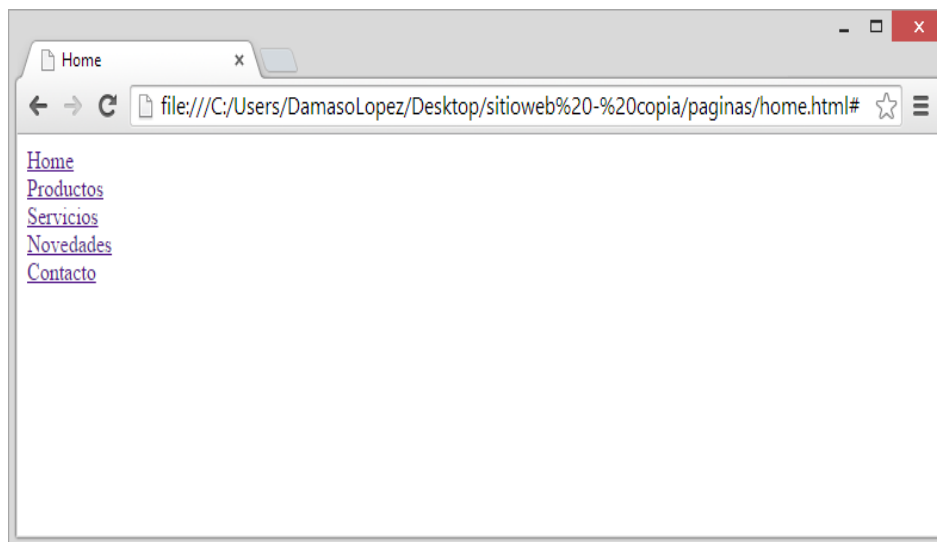




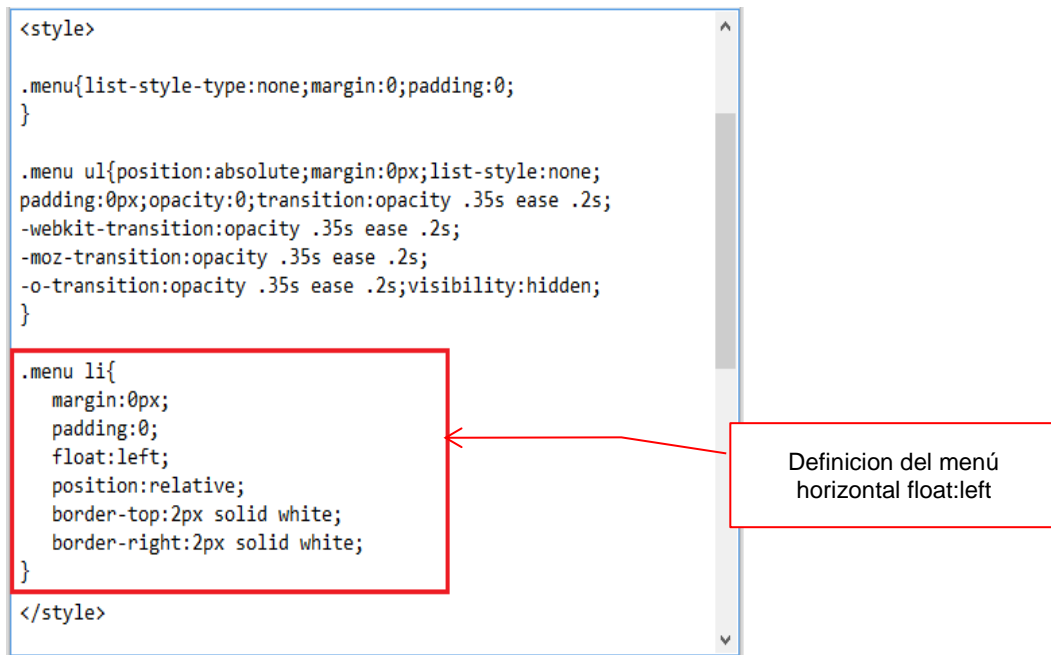
En las reglas de estilo creamos la clase menu. Defina las propiedades para quitar las viñetas y márgenes de la lista en la etiqueta <ul> a través de la propiedad list-style-type: none. Para ocultar los sub menú (.menu ul) definimos la propiedad visibility: hidden, position, márgenes y relleno.



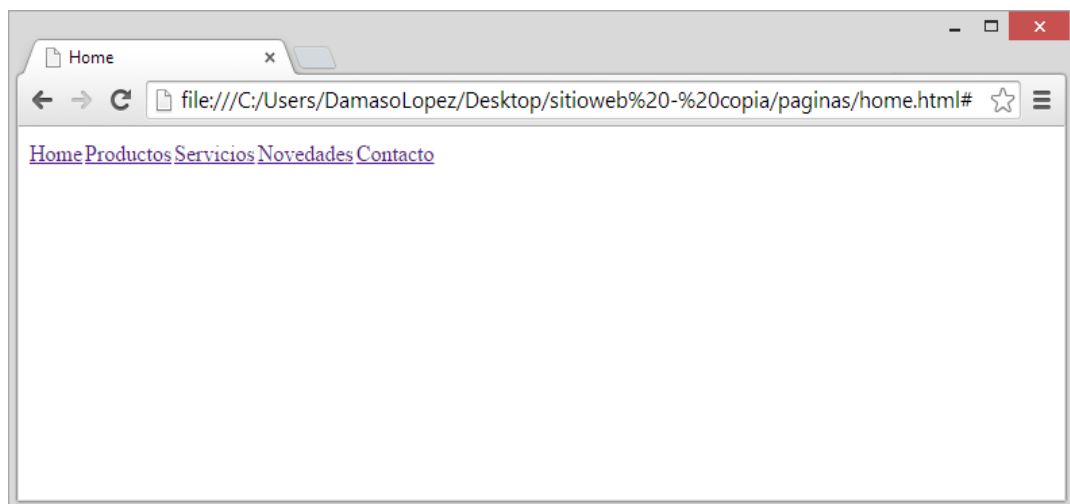
Actualizar el documento en el navegador Chrome, visualizando la página tal como se muestra.



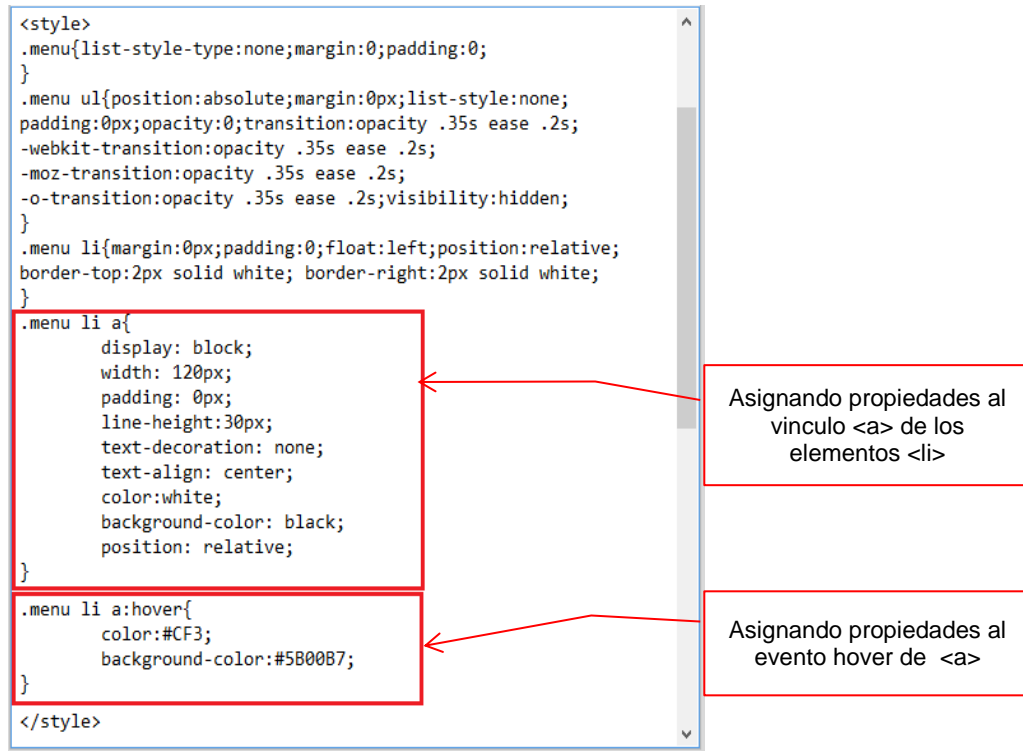
Para que el menú sea horizontal, en la etiqueta <li> de la lista, agregamos la propiedad float: left, para hacer que los elementos de la lista se coloquen uno al lado del otro. También definimos los bordes, posición, márgenes y relleno.



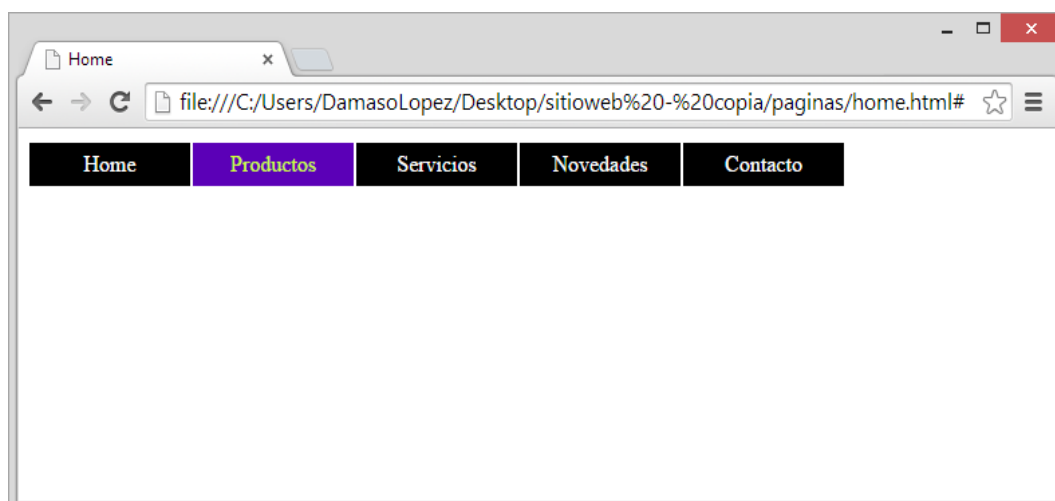
Actualizar el documento en el navegador Chrome, visualizando la página tal como se muestra.



A continuación, definimos estilos directamente sobre los vínculos, <a> asignando: bloque, tamaño, tipo de letra, colores, decoración, etc. Para seleccionar los vínculos dentro de los elementos de lista, encadenamos los nombres de las etiquetas en el selector CSS como lo hicimos con los <li>. Por último, definimos propiedades del evento hover de la etiqueta <a>, para cambiar el color de letra y de fondo al acercar el cursor en el vínculo



Actualizar el documento en el navegador Chrome, visualizando la página tal como se muestra



Por último, definimos propiedades del evento hover de la etiqueta <li> para cada uno de los subitems, donde al pasar el cursor sobre el elemento del menú, visualizamos los elementos del subitem

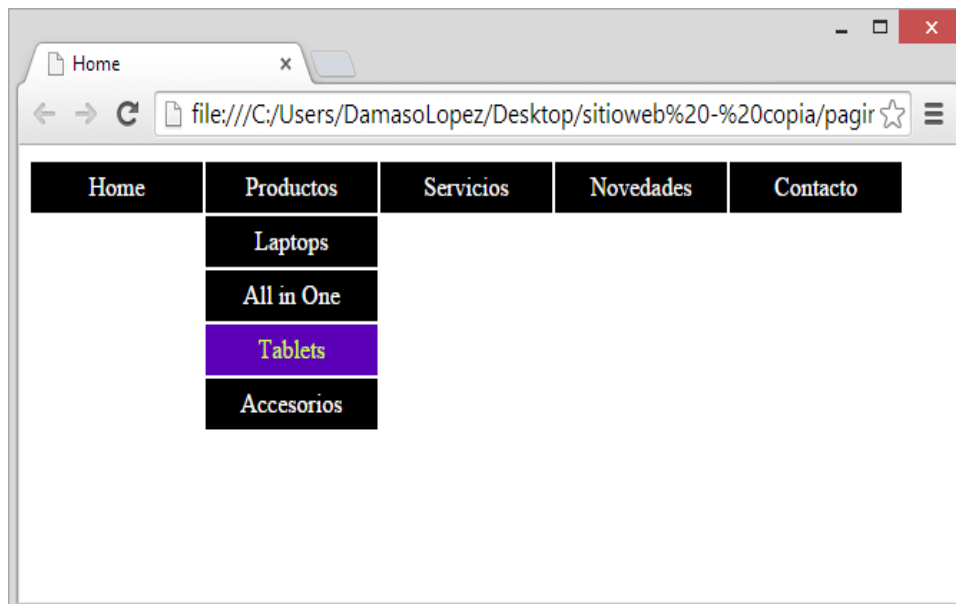
```
<style>
.menu{list-style-type:none;margin:0;padding:0;}
.menu ul{position:absolute;margin:0px;list-style:none;
padding:0px;opacity:0;transition:opacity .35s ease .2s;
-webkit-transition:opacity .35s ease .2s;
-moz-transition:opacity .35s ease .2s;
-o-transition:opacity .35s ease .2s;visibility:hidden;
}
.menu li{margin:0px;padding:0;float:left;position:relative;
border-top:2px solid white; border-right:2px solid white;
}
.menu li a{display: block;width: 120px;padding: 0px;
line-height:30px;text-decoration: none;text-align: center;
color:white;background-color: black;position: relative;
}
.menu li a:hover{color:#CF3;background-color:#5B00B7;}

.menu li:hover ul{
    opacity:1;
    visibility:visible;
}

</style>
```

Cuando el cursor esté encima de un elemento, visualiza sus subitems

Actualizar el documento en el navegador Chrome, visualizando la página tal como se muestra



## LABORATORIO 6.1

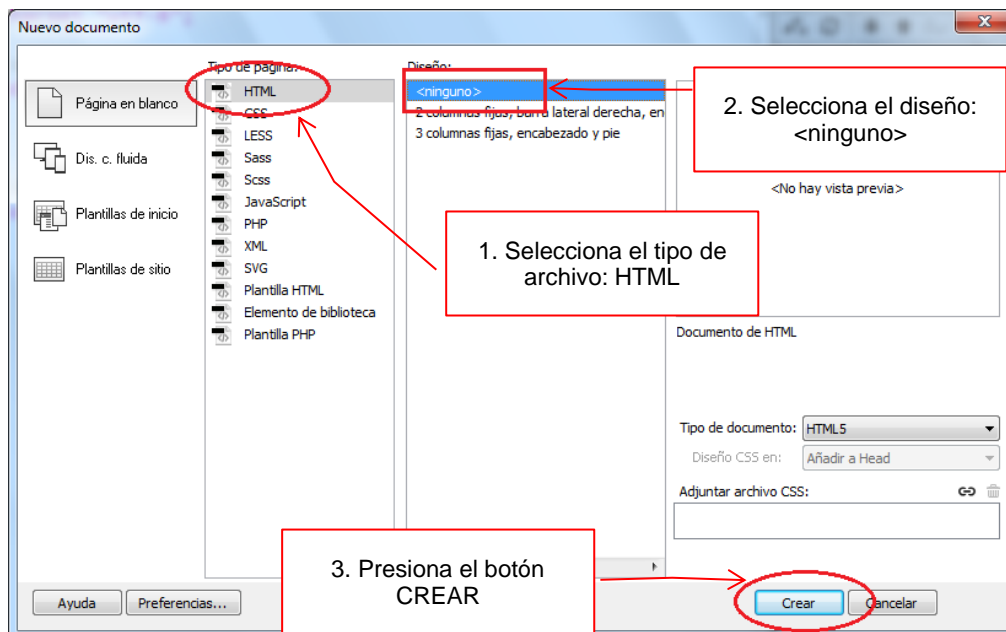
Se nos pide diseñar una página de inicio de la empresa Toshiba, donde implemente bloques de capas, menú de navegación y hojas de estilo CSS.



## Agregando la pagina HTML

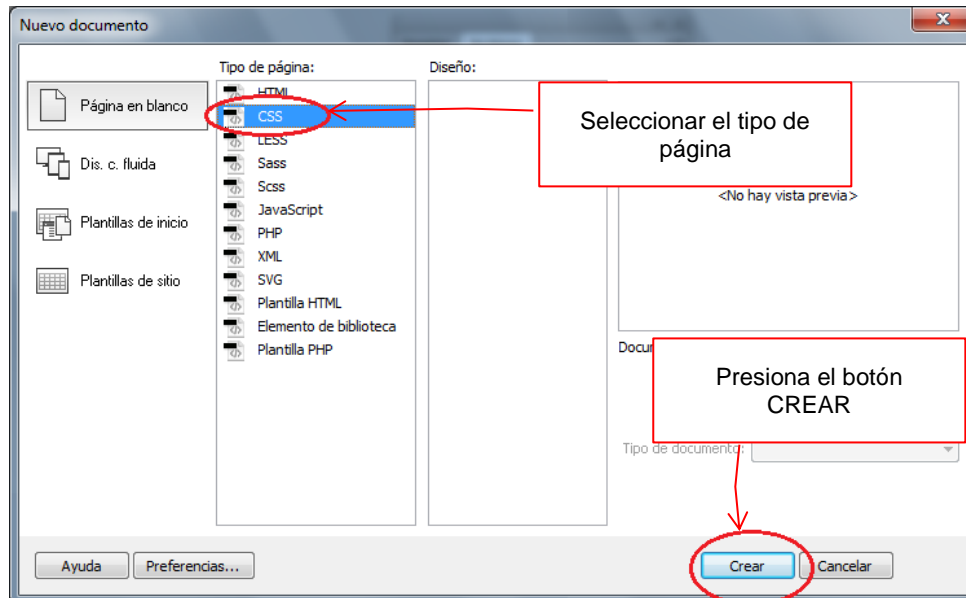
A continuación, creamos un archivo HTML, el cual no tendrá diseño (ninguno), tal como se muestra.

Guardar la página con el nombre de home.html, en la carpeta paginas.

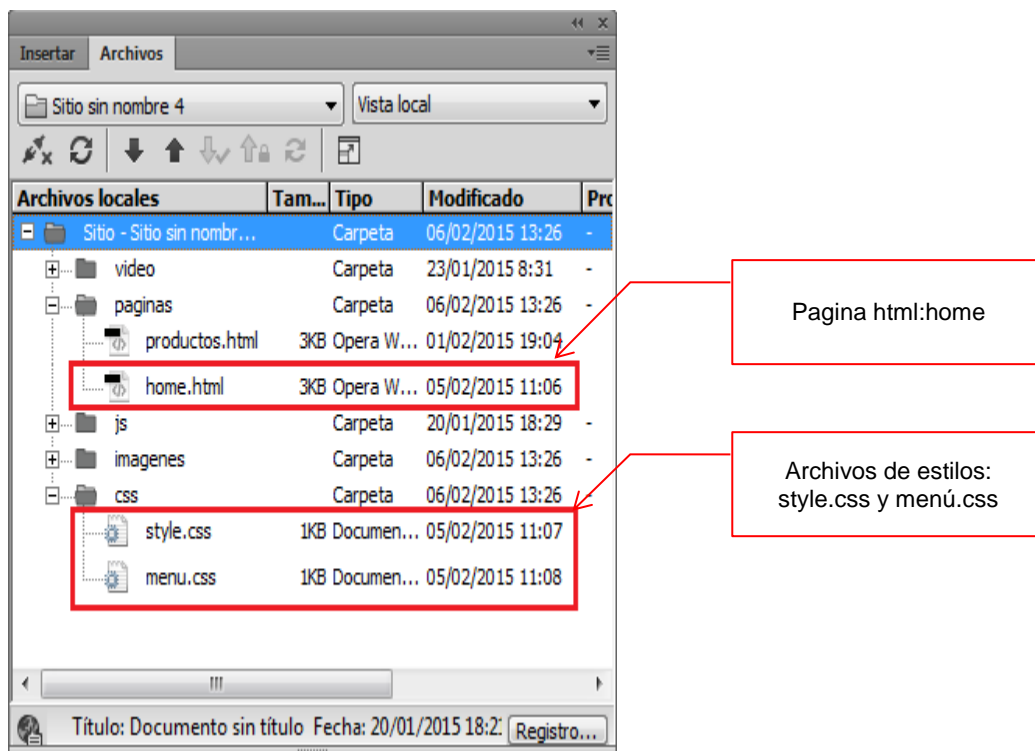


## Agregando un archivo CSS

Antes de iniciar con el diseño de la página HTML, agregamos dos archivos CSS, para definir las reglas de estilos que utilizará la página. Los archivos se llamarán: style.css y menú.css. Creado cada archivo, procedemos a guardarlo en la carpeta css.



Al guardar los archivos estos se visualizan en la carpeta del Sitio Web de esta manera.



## Enlazando las hojas de estilo a la página HTML

En la página home, enlazamos los archivos de estilo a través de la etiqueta <link> el cual hace una referencia a los archivos style.css y menú.css, tal como se muestra.

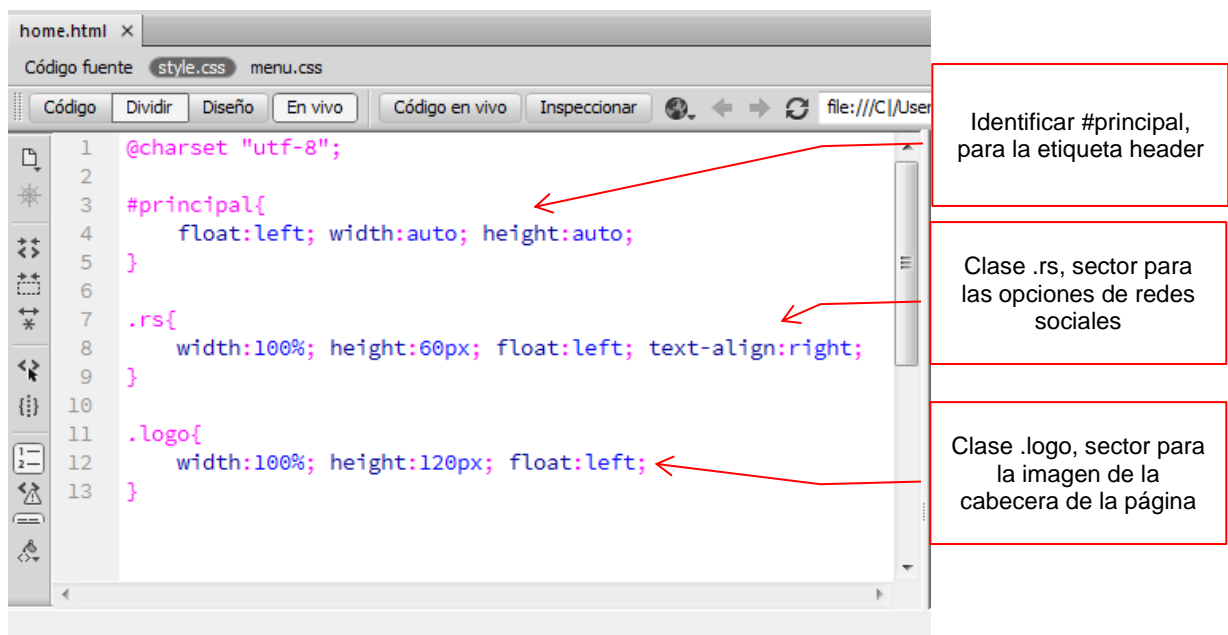


## Agregando reglas de estilo a la cabecera de la página

En el archivo style.css, defines las reglas de estilo para la cabecera:

- #principal, identificador que define las reglas para la etiqueta <header>
- .rs, clase que define el sector para las opciones de redes sociales
- .logo, clase que define el sector para la imagen de la página

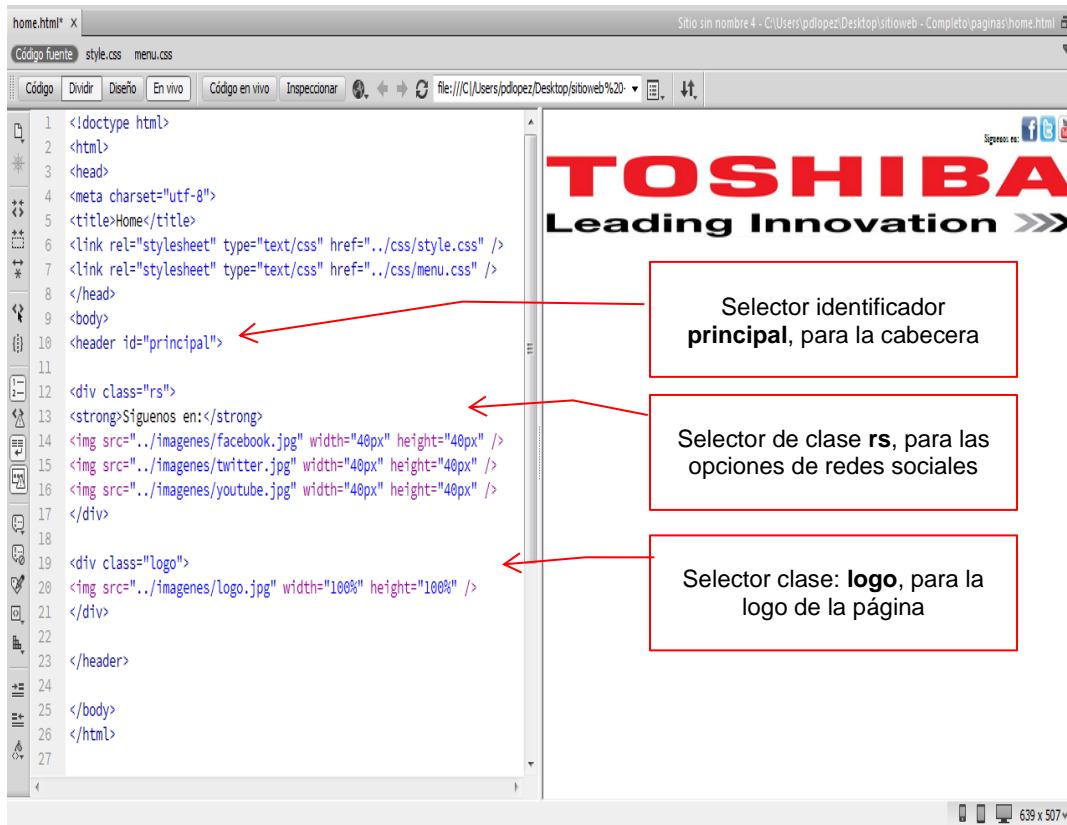
A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <header> y su contenido en la pagina HTML

Definida las reglas en style.css, dentro del body defina la etiqueta <header> asignando el identificador **principal**. Agrega dos bloques div:

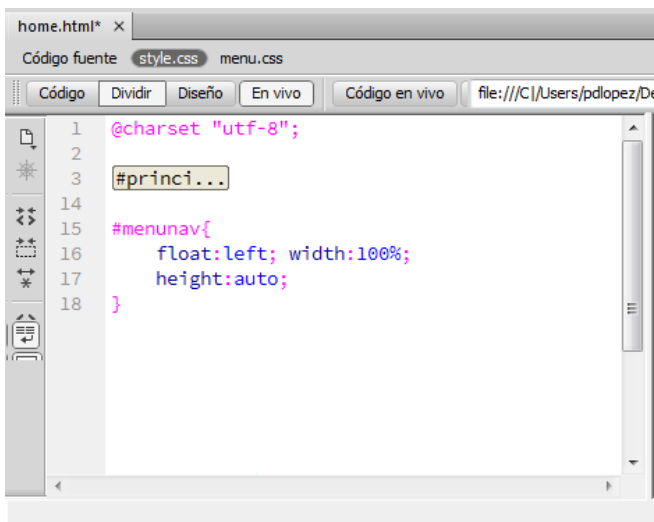
- <div class="rs"> sector para las opciones de redes sociales
- <div class="logo"> sector para el logo de la pagina.



## Agregando reglas de estilo al menú de navegación <nav>

En el archivo style.css, definas las reglas de estilo para el menú de navegación:

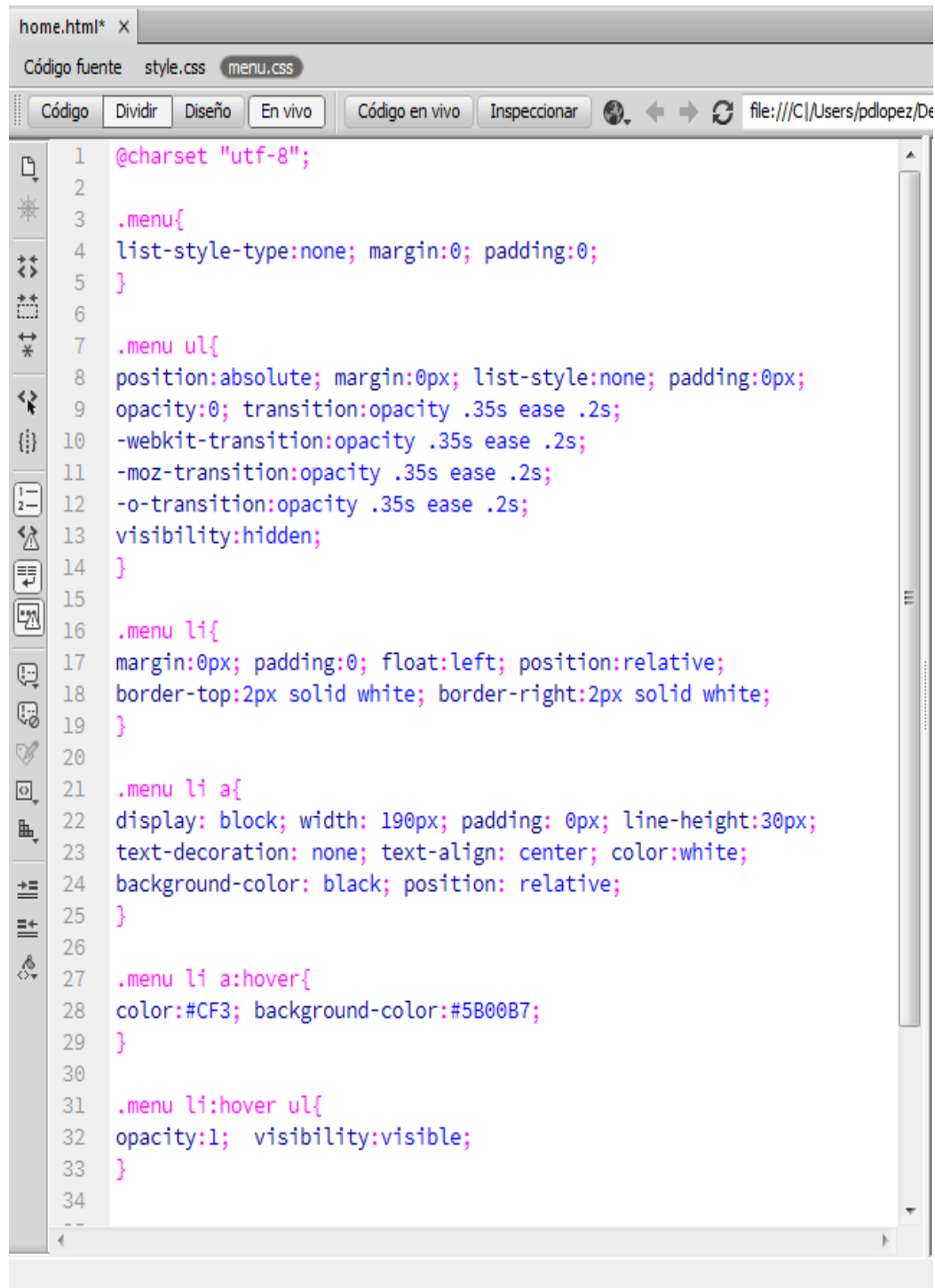
- #menunav, identificador que define las reglas para el bloque <nav>





## Definiendo las reglas para el archivo menú.css

En el archivo menú.css defina las reglas para crear un menú desplegable.

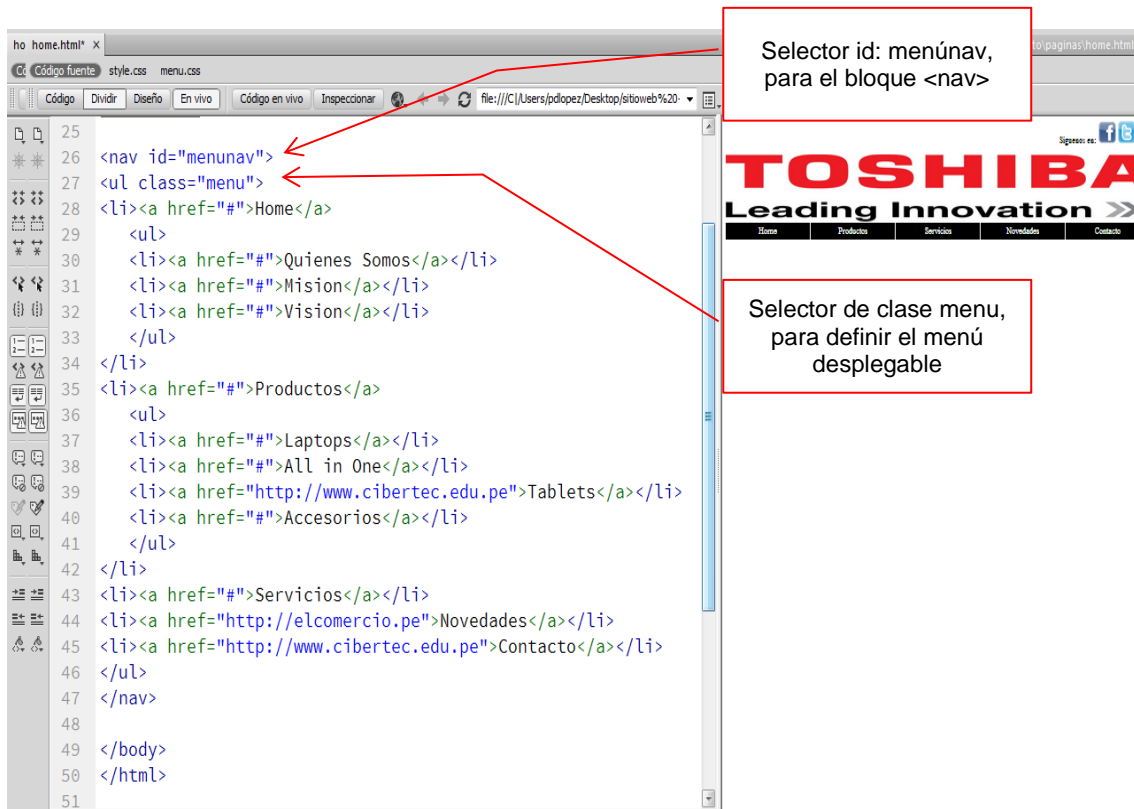


```
home.html* x
Código fuente style.css menu.css
Código Dividir Diseño En vivo Código en vivo Inspeccionar file:///C:/Users/pdlopez/De

1 @charset "utf-8";
2
3 .menu{
4   list-style-type:none; margin:0; padding:0;
5 }
6
7 .menu ul{
8   position:absolute; margin:0px; list-style:none; padding:0px;
9   opacity:0; transition:opacity .35s ease .2s;
10  -webkit-transition:opacity .35s ease .2s;
11  -moz-transition:opacity .35s ease .2s;
12  -o-transition:opacity .35s ease .2s;
13  visibility:hidden;
14 }
15
16 .menu li{
17   margin:0px; padding:0; float:left; position:relative;
18   border-top:2px solid white; border-right:2px solid white;
19 }
20
21 .menu li a{
22   display: block; width: 190px; padding: 0px; line-height:30px;
23   text-decoration: none; text-align: center; color:white;
24   background-color: black; position: relative;
25 }
26
27 .menu li a:hover{
28   color:#CF3; background-color:#5B00B7;
29 }
30
31 .menu li:hover ul{
32   opacity:1; visibility:visible;
33 }
34
```

## Agregando la etiqueta <nav> y sus enlaces en la página HTML

Definida las reglas en los archivos de estilos, defina la etiqueta <nav> agregando una lista no ordenada, sus ítems y enlaces <a href>. Presiona la tecla F5 para visualizar el diseño de la página.

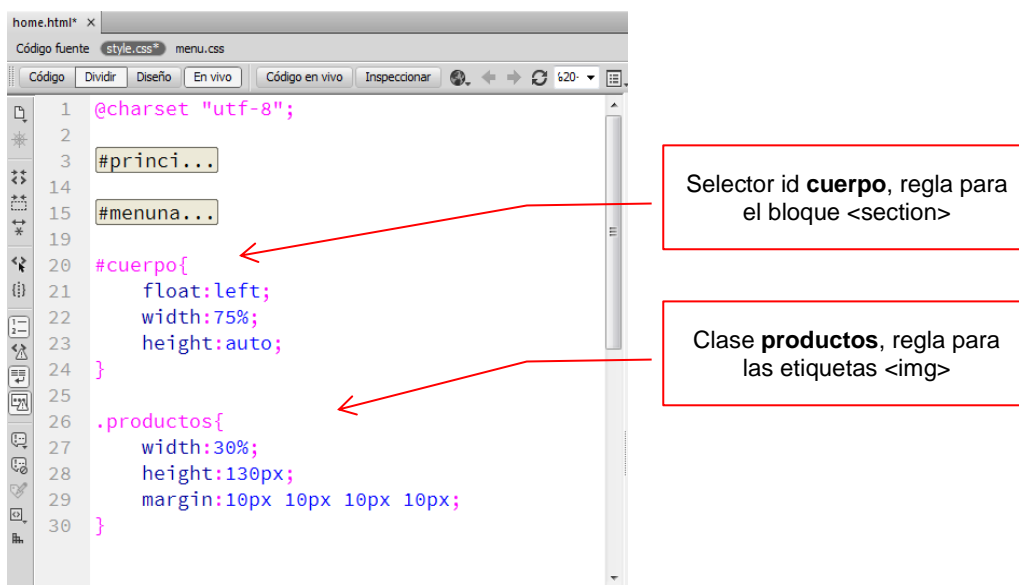


## Agregando las reglas en el cuerpo de la pagina <section>

En el archivo CSS, style-productos.css defina las reglas para el cuerpo de la página

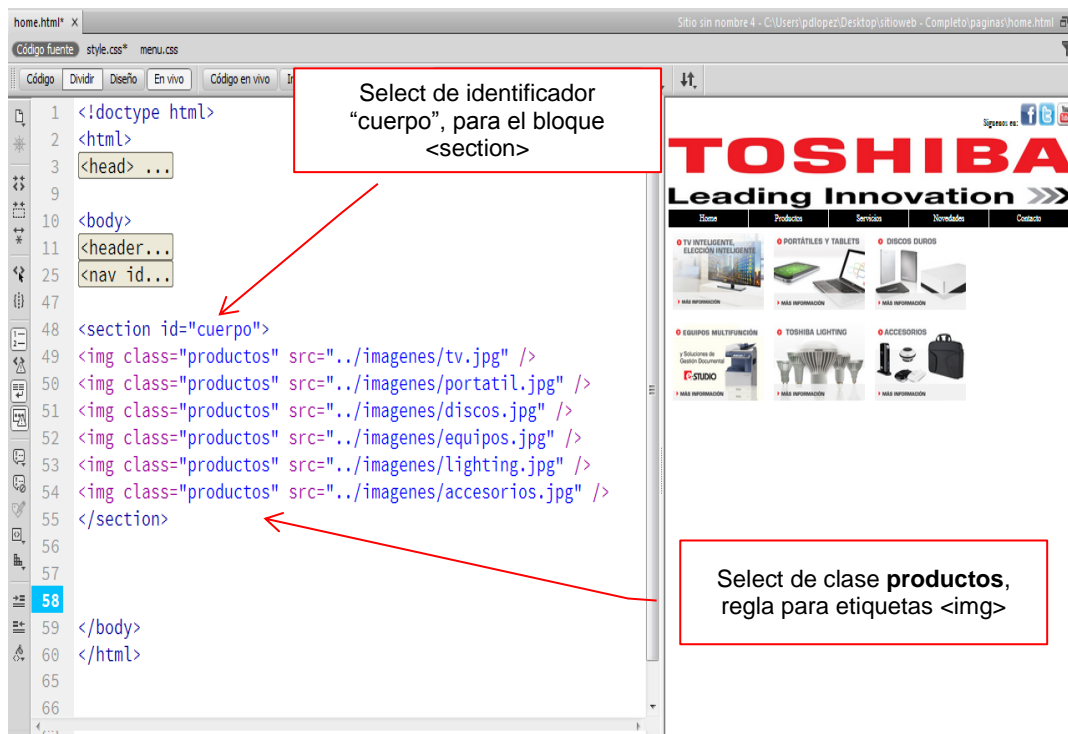
- #cuerpo, regla para el bloque <section>, su ancho será 75% de la página
- .producto, regla para las imágenes

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <section> para la página HTML

Definida las reglas, define la etiqueta <section> para promocionar los productos. Presiona la tecla F5 para visualizar el diseño

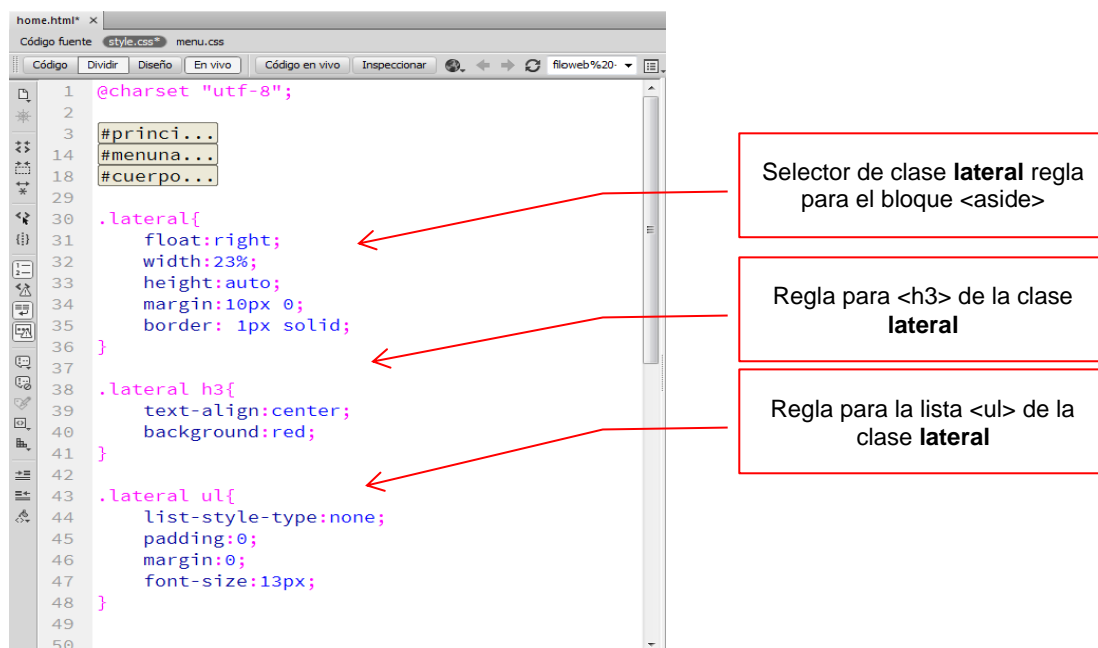


## Agregando las reglas en la barra lateral <aside>

En el archivo CSS, style-productos.css define las reglas para barra lateral.

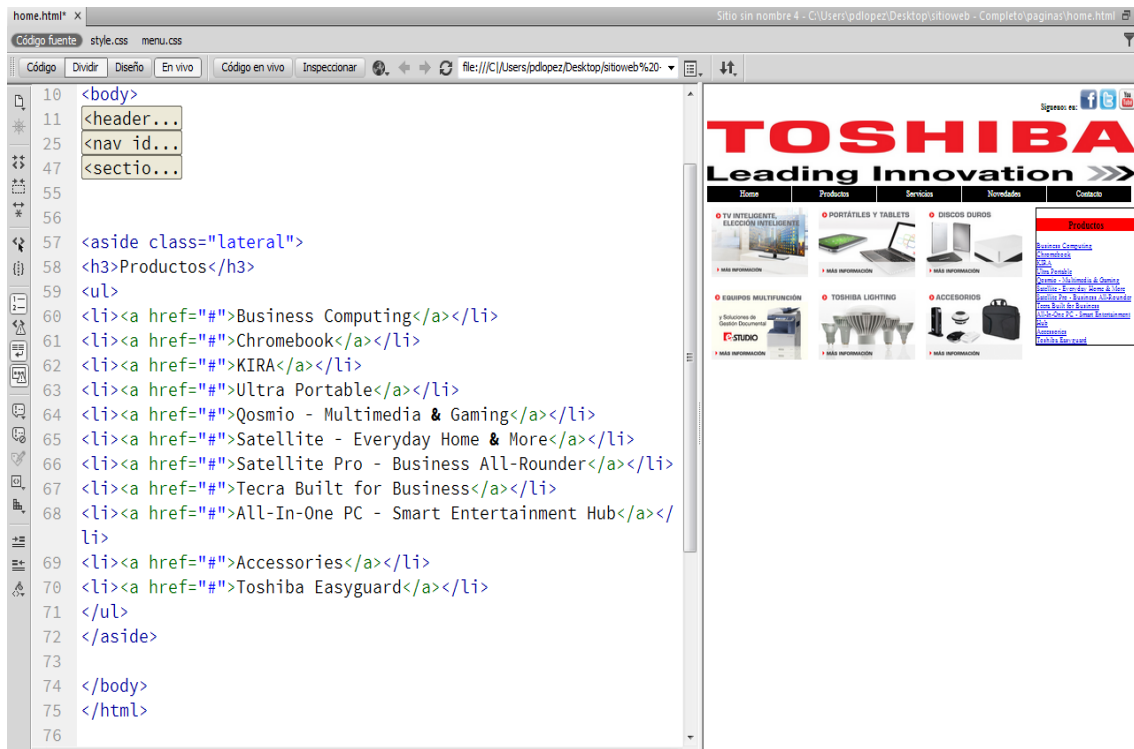
- .lateral, regla para el bloque <aside> donde su posición flotante es derecha
- .lateral h3, regla para la etiqueta <h3> del bloque <aside>
- .lateral ul, regla para la lista <ul> dentro de cada bloque article

A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <aside> para la página HTML

Definida las reglas en el archivo css, defina la etiqueta <aside> para mostrar información adicional de los productos. Presiona la tecla F5 para visualizar el diseño

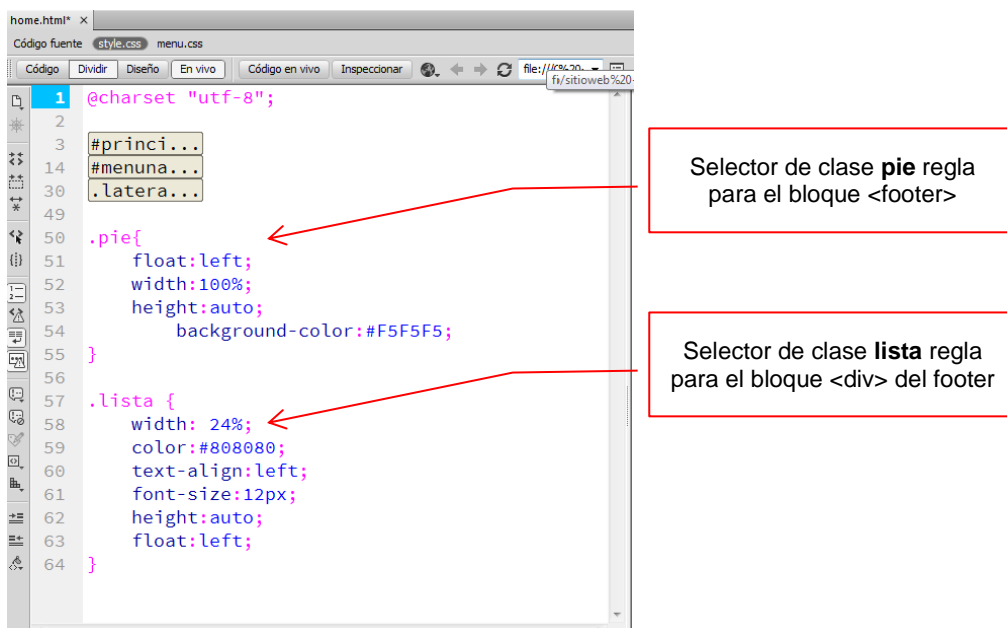


## Agregando las reglas al pie de página

En el archivo CSS, style.css definimos las reglas para el pie de página.

- #pie, clase para el bloque <footer>
- .lista, clase para definir una regla del bloque div del footer, lista de productos

A continuación, guardar el archivo estilo.css: CTRL+ S



## Agregando el pie de página <footer>

Definida las reglas en style.css, defina la etiqueta <footer> agregando su identificador **pie**, tal como se muestra. Presiona la tecla F5 para visualizar el diseño

Selector de clase: pie, asignando a <footer>

```

72 <footer class="pie">
73 <div class="lista">
74 Satellite<br/>      Satellite Pro<br/>      Tecra<br/>
75 Tablet<br/>      Accesorios
76 </div>
77
78 <div class="lista">
79 Led 3D TV<br/>      Smart TV<br/>      LED TV<br/>
80 Combi TV<br/>      Reproductores<br/>      Accesorios TV
81 </div>
82
83 <div class="lista">
84 Impresoras, Copiadoras, Fax<br/>
85 Sistemas Medicos<br/>      Comunicaciones<br/>
86 Visualizacion Avanzada<br/>      Sondas Medicas<br/>
87 Aire Acondicionado<br/>      Memoria
88 </div>
89
90 <div class="lista">
91 Soporte TV<br/>      Portatiles<br/>      Registros<br/>
92 Servicios
93 </div>
94
95 <div class="pie">
96 
97 </div>
98 </footer>
  
```

Selector de clase: lista, para el bloque <div> donde lista los productos de la empresa

Selector de clase: pie, para el bloque <div> de la imagen del pie de la página

Para ejecutar a través del navegador, primero guardar la página: CTRL+S y presiona la tecla F12 para ejecutar la página con el navegador.

## Resumen

- 📖 HTML5 y CSS3 son dos recursos esenciales para diseñadores y desarrolladores en la actualidad si lo que se busca es crear sitios web (funcionales) así como estéticamente hermosos. Estos lenguajes permiten crear páginas web con efectos realmente alucinantes e innovadores y además son muy simples de usar.
- 📖 Otro elemento bastante típico y común en muchas páginas web es la zona de navegación, formada por enlaces a las zonas principales de la web, o por un menú de navegación, desplegable de enlaces, etc. Por lo visto Html5 interpreta que es una zona que deberíamos identificar y ha creado para ello la nueva etiqueta **<nav>** **</nav>**.
- 📖 Para diseñar un menú de navegación hacemos uso de listas, porque un menú básicamente es una lista, no es un párrafo, ni un título, es una lista de vínculos.
- 📖 Para cambiar el aspecto de las listas, CSS nos permite realizar cambiar el aspecto de la lista, para convertirla en el bonito menú que queremos, sin que deje de ser lista. Pero primero, veamos en qué consisten los pre-formatos que trae de por sí una lista.
- 📖 Las listas se renderizan en el navegador con ciertas características, como el margen/relleno (los cuales son interpretados de diferente manera en los distintos navegadores como opera, Chrome, Internet, entre otros) de la lista completa, el margen / relleno (idem) de cada ítem de la lista y las viñetas, además de un salto de línea para cada ítem por ser elementos de bloque.
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 <http://www.cristalab.com/tutoriales/crear-menus-de-navegacion-en-css-usando-listas-c130/>

🔗 <http://html5facil.com/tutoriales/como-crear-menu-basico-html5-css3/>

🔗 <http://xn--diseandoweb-4db.com/htmlcss/como-crear-un-menu-con-html-5-y-css/>

🔗 <http://www.cssblog.es/menu-con-varios-niveles-mediante-css/>

🔗 <http://www.antocas.com/menu-desplegable-basico-html5-y-css3-drop-down-submenu/>

🔗 <http://ivanprego.com/disenio-web/css/crear-un-menu-estilo-apple-con-html5-y-css3/>

🔗 <http://extremisimo.com/29-menus-desplegables-en-html5-y-css3-gratuitos/>



# MULTIMEDIA

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3 y multimedia.

## TEMARIO

### 4.1 Tema 7: Elementos multimedia

- 4.1.1 Etiquetas AUDIO y VIDEO
- 4.1.2 Codificar audio y video para la web
  - 4.1.2.1 Autoplay, loop, preload, controles (audio)
  - 4.1.2.2 Controles, width/height, poster (video)
  - 4.1.2.3 Filetypes (source) Múltiples formatos de audio y video para todos los diversos navegadores.
- 4.1.3 Convertidor de audio y video online: mp3, ogg, mp4
- 4.1.4 Etiqueta iframe: Mostrar video de Youtube
- 4.1.5 Uso de la herramienta Lightbox para la gestión de imágenes y video.

## ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5, hojas de estilo CSS3 y bloques de contenidos
- Los alumnos diseñan un menú en una página web





## 4.1 ELEMENTOS MULTIMEDIA

Algunos de los cambios más notables en HTML5 es la adición de etiquetas para soportar elementos multimedia y gráficos dinámicos. Esto incluye la etiqueta **<audio>** para reproducir sonidos, **<video>** para videos y **<canvas>** para gráficos dinámicos, todo esto dentro del web browser y sin depender de software externo. Los nuevos elementos multimedia habilitan a los browsers modernos para proveer experiencias enriquecidas y gráficos dinámicos a partir del código HTML.

El soporte para usar video, manipular imágenes, y visualización de datos permitirá a los diseñadores y desarrolladores crear experiencias ricas y atractivas que funcionen en múltiples plataformas.

### 4.1.1 Etiquetas AUDIO y VIDEO

Hasta ahora, para incluir un elemento multimedia en un documento, se hacía uso del elemento **<object>**, cuya función es incluir un elemento externo genérico. Debido a la incompatibilidad entre navegadores, se hacía también necesario el uso del elemento **<embed>** y duplicar una serie de parámetros.

HTML5 introduce soporte integrado para el contenido multimedia gracias a los elementos **<audio>** y **<video>**, ofreciendo la posibilidad de insertar contenido multimedia en documentos HTML.

### 4.1.2 Codificar audio y video para la web

A continuación, presentamos un ejemplo para reproducir el video abc.mp4 en un documento HTML.

```
<video src="../../videos/abc.mp4" controls>
</video>
```

En este ejemplo insertamos el audio abc.ogg en un documento HTML.

```
<audio src="../../videos/abc.ogg" controls>
</audio>
```

En estos dos ejemplos, los elementos **<audio>** y **<video>** define el origen del archivo multimedia a través de la propiedad **src**. Este atributo especifica la URL del archivo a ser reproducido. Al igual que en el elemento **<video>** normalmente será reemplazado por el elemento **<source>** para ofrecer diferentes formatos de audio entre los que el navegador pueda elegir.

#### 4.1.2.1 Autoplay, loop, preload, controles (audio)

El elemento **<audio>** trabaja del mismo modo y comparte varios atributos con el elemento **<video>**:

- **src**. Este atributo especifica la URL del archivo a ser reproducido. Al igual que en el elemento **<video>** normalmente será reemplazado por el elemento **<source>** para ofrecer diferentes formatos de audio entre los que el navegador pueda elegir.

- **controls.** Este atributo activa la interface que cada navegador provee por defecto para controlar la reproducción del audio.
- **autoplay.** Cuando este atributo está presente, el audio comenzará a reproducirse automáticamente tan pronto como sea posible.
- **loop.** Si este atributo es especificado, el navegador reproducirá el audio una y otra vez de forma automática.
- **preload.** Este atributo puede tomar tres valores diferentes: none, metadata o auto. El primero indica que el audio no debería ser cacheado, normalmente con el propósito de minimizar tráfico innecesario. El segundo valor, metadata, recomendará al navegador obtener información sobre el medio (por ejemplo, la duración). El tercer valor, auto, es el valor configurado por defecto y le aconseja al navegador descargar el archivo tan pronto como sea posible.

En el siguiente ejemplo vamos a reproducir el audio en el documento HTML, donde se muestre los controles del reproductor, se reproduzca y se repita automáticamente

```
<audio src="../audio/abc.ogg" controls autoplay loop >
<p>Tu navegador no implementa el Audio</p>
</audio>
```

#### 4.1.2.2 Controles, width/height, poster (video)

Entre los atributos de la etiqueta **<video>**:

- **poster.** El atributo poster indica la imagen que el navegador debe mostrar mientras el vídeo se está descargando, o hasta que el usuario reproduce el vídeo. Si no se indica este atributo, el navegador muestra el primer fotograma del vídeo, que puede no ser representativo del vídeo que se va a reproducir.
- **height, width:** Los atributos height y width indican al navegador el tamaño del vídeo en pixels. Si no se indican estas medidas, el navegador utiliza las medidas definidas en el vídeo de origen, si están disponibles. De lo contrario, utiliza las medidas definidas en el fotograma poster, si están disponibles. Si ninguna de estas medidas está disponible, el ancho por defecto es de 300 pixels. Si únicamente se especifica una de las dos medidas, el navegador automáticamente ajusta la medida de la dimensión no proporcionada, conservando la proporción del vídeo. Si por el contrario, se especifican las dos medidas, pero no coinciden con la proporción del vídeo original, el vídeo no se deforma a estas nuevas dimensiones, sino que se muestra en formato letterbox manteniendo la proporción original.

En el siguiente ejemplo vamos a reproducir un video en el documento HTML, donde se muestre los controles del reproductor y cuyas dimensiones del reproductor es de ancho 200px y alto 120 px.

```
<video src="../video/abc.m4" controls width="200px"
height="120px">
<p>Tu navegador no implementa el Video</p>
</video>
```

#### 4.1.2.3 Filetypes (source) Múltiples formatos de audio y video para todos los diversos navegadores.

No existe un estándar para formatos de video y audio en la web. Existen varios contenedores y diferentes codificadores disponibles, pero ninguno fue totalmente adoptado y no hay consenso alguno de parte de los fabricantes de navegadores para lograr un estándar en el futuro cercano.

Los contenedores más comunes son OGG, MP4, FLV y el nuevo propuesto por Google, WEBM. Normalmente estos contenedores contienen video codificado con los codificadores Theora, H.264, VP6 o VP8, respectivamente. Esta es la lista de los más usados:

- OGG codificador de video Theora y audio Vorbis.
- MP4 codificador de video H.264 y audio AAC.
- FLV codificador de video VP6 y audio MP3. También soporta H.264 y AAC.
- WEBM codificador de video VP8 y audio Vorbis.

Los codificadores utilizados para OGG y WEBM son gratuitos, pero los utilizados para MP4 y FLV están patentados, lo que significa que si queremos usar MP4 o FLV para nuestras aplicaciones deberemos pagar. Algunas restricciones son anuladas para aplicaciones gratuitas.

El tema es que en este momento Safari e Internet Explorer no soportan la tecnología gratuita. Ambos solo trabajan con MP4 y solo Internet Explorer anunció la inclusión del codificador VP8 en el futuro. Esta es la lista de los navegadores más populares:

- Firefox codificador de video Theora y audio Vorbis.
- Google Chrome codificador de video Theora y audio Vorbis. También soporta codificador de video H.264 y audio AAC.
- Opera codificador de video Theora y audio Vorbis.
- Safari codificador de video H.264 y audio AAC.
- Internet Explorer codificador de video H.264 y audio AAC.

Un mayor soporte para el formato WEBM en el futuro podría mejorar la situación, pero probablemente no habrá un formato estándar por al menos los próximos dos o tres años y tendremos que considerar diferentes alternativas de acuerdo a la naturaleza de nuestra aplicación y nuestro negocio

Una vez que disponemos del vídeo o el audio en los distintos formatos, es necesario indicar todas las ubicaciones de estos formatos, para que sea el navegador el que decida que formato reproducir. Evidentemente, no podemos especificarlos todos dentro del atributo src, por lo que tendremos que hacerlo de manera separada utilizando el elemento **<source>**.

```
<video controls>
<source src="abc.mp4" type="video/mp4">
<source src="abc.webm" type="video/webm">
<p>Tu navegador no puede soportar el video.
Descargar el video en el formato
<a href="abc.webm">webM</a> o <a href="abc.mp4">MP4</a>
</p>
</video>
```

#### 4.1.3 Convertidor de audio y video online: mp3, ogg, mp4

Un desarrollador, si desea implementar páginas HTML con reproductores de video, va a tener que codificarse al menos en dos (idealmente en tres) formatos de vídeo para que puedan funcionar con la especificación de HTML5. Actualmente se dispone de varias herramientas que nos pueden ayudar con esta labor.

- **Miro Video Converter:** Es tu herramienta de referencia: este conversor de código abierto, disponible en versiones para Mac y para Windows, convierte prácticamente cualquier formato de video de origen a todos los formatos de HTML5 que quieras utilizar y también a otros formatos específicos de dispositivos y plataformas.
- **Handbrake:** Otro buen conversor de código abierto para Mac y Windows (y Linux también). Handbrake es una buena opción concretamente para convertir a H.264.
- **MPEG Streamclip ():** Aunque no es código abierto, MPEG Streamclip se puede conseguir gratuitamente en versiones para Mac y Windows. Se maneja también con H.264, aunque puede convertir vídeos a otros muchos formatos que nos pueden servir para utilizarlos fuera de la especificación de HTML5.
- **Firefogg:** Es una herramienta más especializada. Firefogg es una extensión de Firefox que convierte videos únicamente a formato Ogg, de ahí el nombre. Para poder utilizarla necesitarás descargar antes Firefox y luego instalar esta extensión.

#### 4.1.4 Etiqueta iframe: Mostrar video de Youtube

El elemento HTML inline frame (<iframe>) representa un contexto de navegación anidada, el cual permite incorporar otra página HTML en la página actual. Cada contexto de navegación tiene su propia historia, sesión y documento activo. El contexto de navegación que incluye el contenido implícito se llama contexto de navegación principal. El contexto de navegación de nivel superior (que no tiene padre) es típicamente la ventana del navegador.

Las propiedades de esta etiqueta en HTML5:

- **src**, URL del documento HTML que se visualiza en el iframe.
- **height**, altura que ocupará el iframe en el documento, se expresa en pixeles.
- **width**, ancho que ocupará el iframe en el documento, se expresa en pixeles.
- **align**, alineamiento del frame en la página.
- **name**, nombre que identifica al iframe.
- **longdesc**, dirección url en la que puede encontrarse una descripción larga del contenido del iframe.
- **frameborder**, especifica el grosor del border del frame, por defecto es cero.
- **framespacing**, especifica el espacio entre la caja y el contenido del frame.
- **scrolling**, "yes | no | auto" - Indica si el iframe debe mostrar barras de scroll (horizontal y vertical) cuando el contenido incluido no cabe en el iframe.

En este ejemplo, la etiqueta html **<iframe>** se utiliza para incrustar un video de youtube.

```
<iframe width="560" height="315" src="//www.youtube.com/
embed/5AKY58ZSJGs" frameborder="0" allowfullscreen="">
</iframe>
```

Para hacer un posicionamiento más específico o darle unas características más concretas a nuestros IFRAMES podemos utilizar la etiqueta style.

#### 4.1.5 Uso de la herramienta Lightbox para la gestión de imágenes y video.

Video Lightbox JS es un programa gratuito que te ayuda a integrar fácilmente vídeo al sitio web, página web o blog, en unos pocos clics sin tener que escribir una sola línea de código.

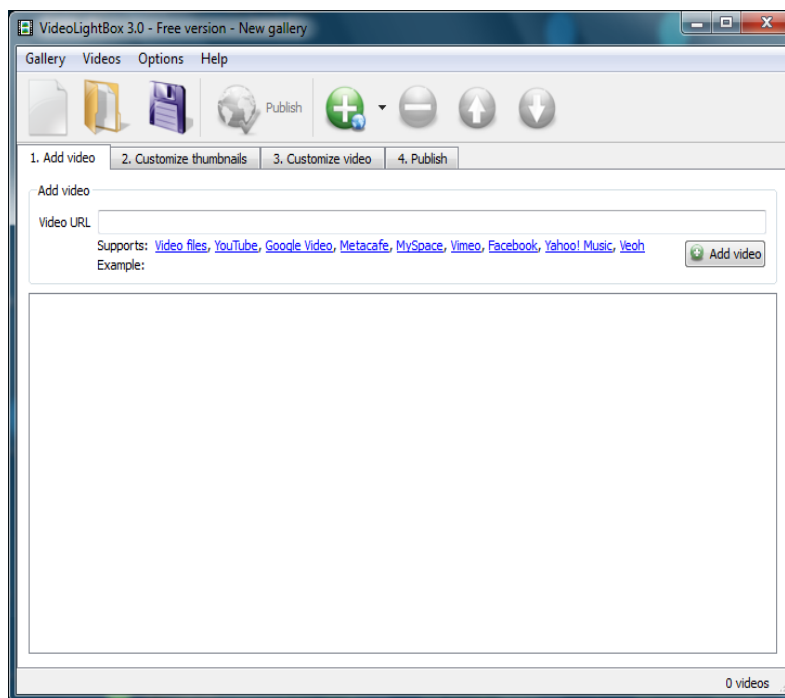
Todo lo que se necesita son 3 pasos sencillos: 1. Incluya Video, 2. Seleccione plantilla, 3. Publique. En el primer paso, incuya un clip de video, por ejemplo, cortar y pegue un URL de YouTube, o arrastre un vídeo de su archivo, a continuación, seleccione la plantilla para el video junto con su imagen representativa y, en el último paso, publique el resultado en un disco local o directamente a Internet a través de un cliente FTP integrado.

Video LightBox genera una imagen representativa para su video clip. Al clicar, esta imagen se expande en una elegante ventana superpuesta que contiene el vídeo. No CSS, HTML, código Javascript, es necesario, ni siquiera la edición de imágenes, solo necesita un par de clics para obtener el video agregado a la Web!

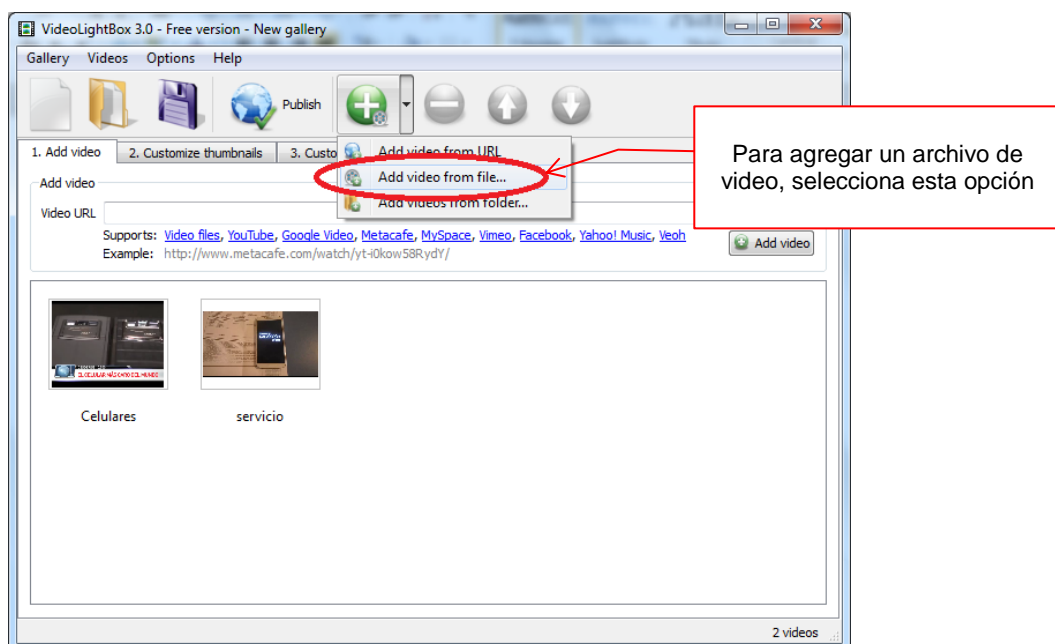
##### Características

- Inserta FLV, MP4, 3GP
- Añade videos de Youtube, Facebook, Google Video, Metacafe, Vimeo, MySpace.
- Versiones de Macintosh y de Windows
- Compatible con HTML
- Creación automática de imágenes
- Adición de título
- Una gran cantidad de temas
- FTP integrado

## Paso 1 - Incluir vídeo a la galería de su sitio web. Abrir el aplicativo VideoLight



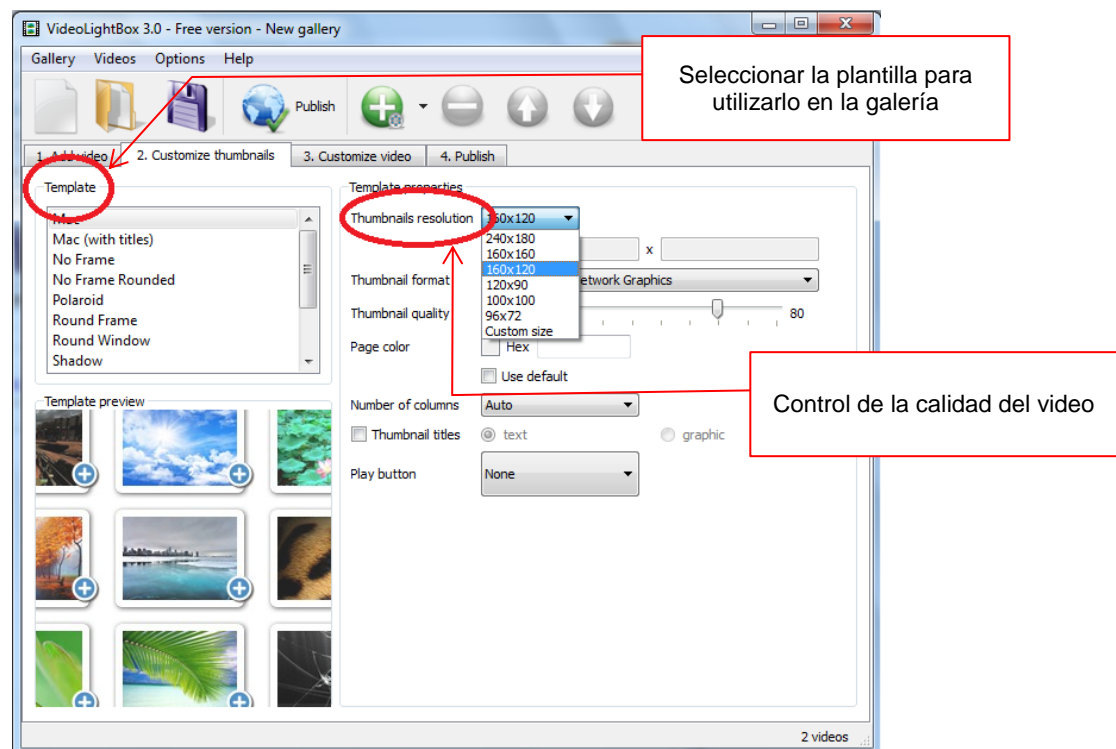
Selecciona la opción Agregar, para agregar video desde YouTube, Archivo. Selecciona los archivos de videos, y se añadiran automaticamente a la lista



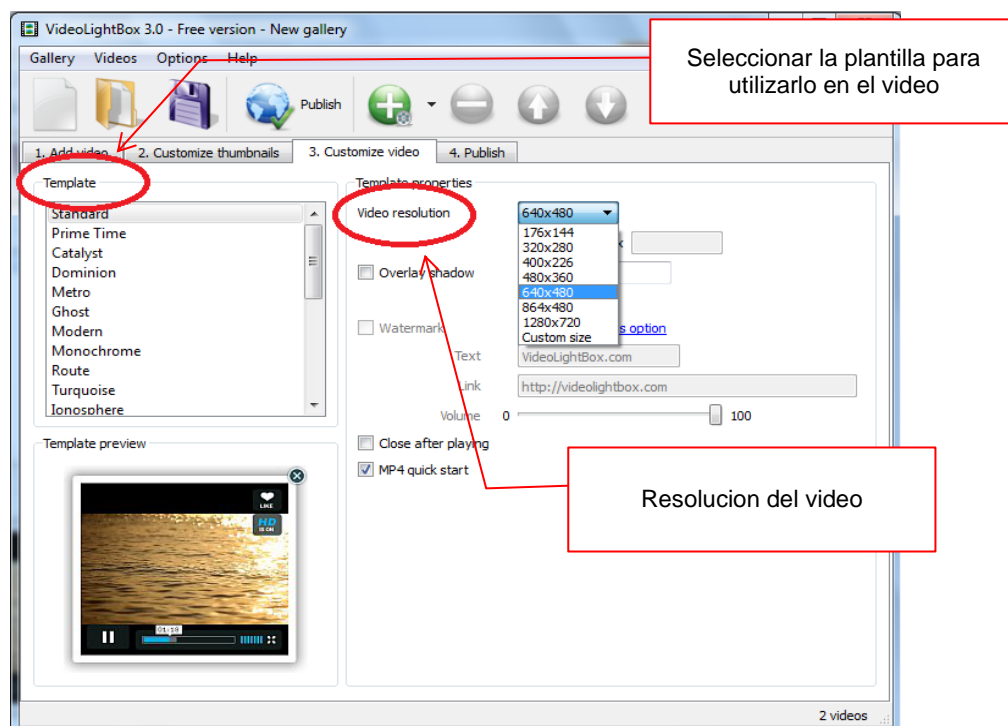
Si ha incluido los videos que usted no desea tener en la galería de sitio web, usted puede facil eliminarlos. Seleccione todos los vídeos que desea eliminar de la galería del sitio web, y seleccione el boton de 'Eliminar archivos seleccionados' de la barra de herramientas. Usted puede escoger y elegir los videos presionando la tecla CTRL mientras hace clic en los videos que quiere.

## Paso 2 - Propiedades de video de la Galeria.

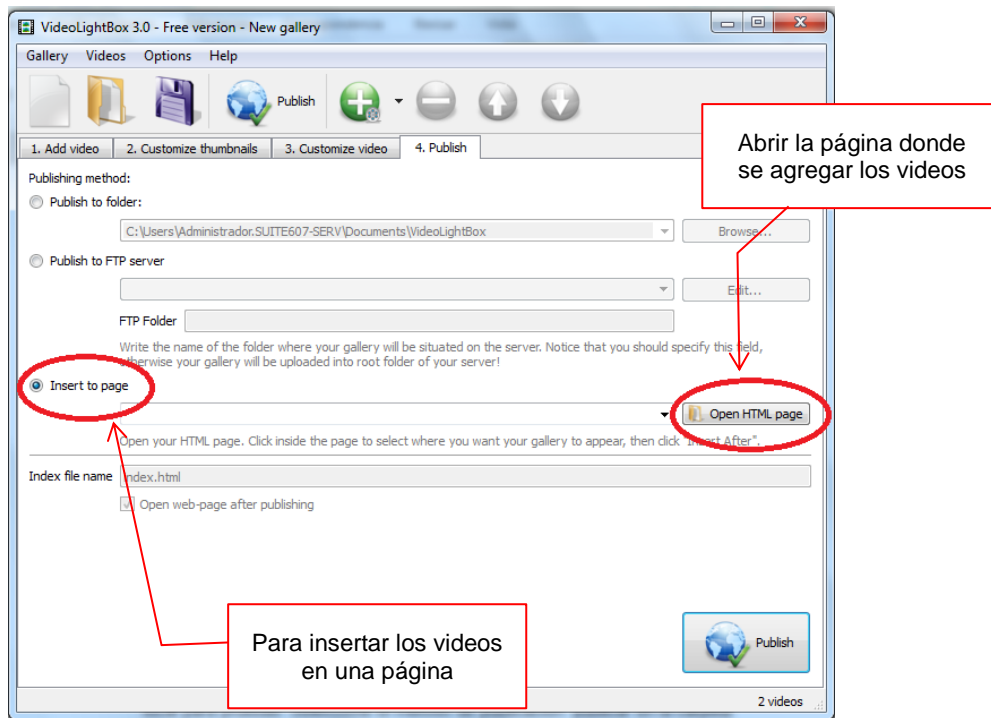
Cambiar entre las plantillas disponibles para seleccionar una plantilla que desea utilizar en la galería.



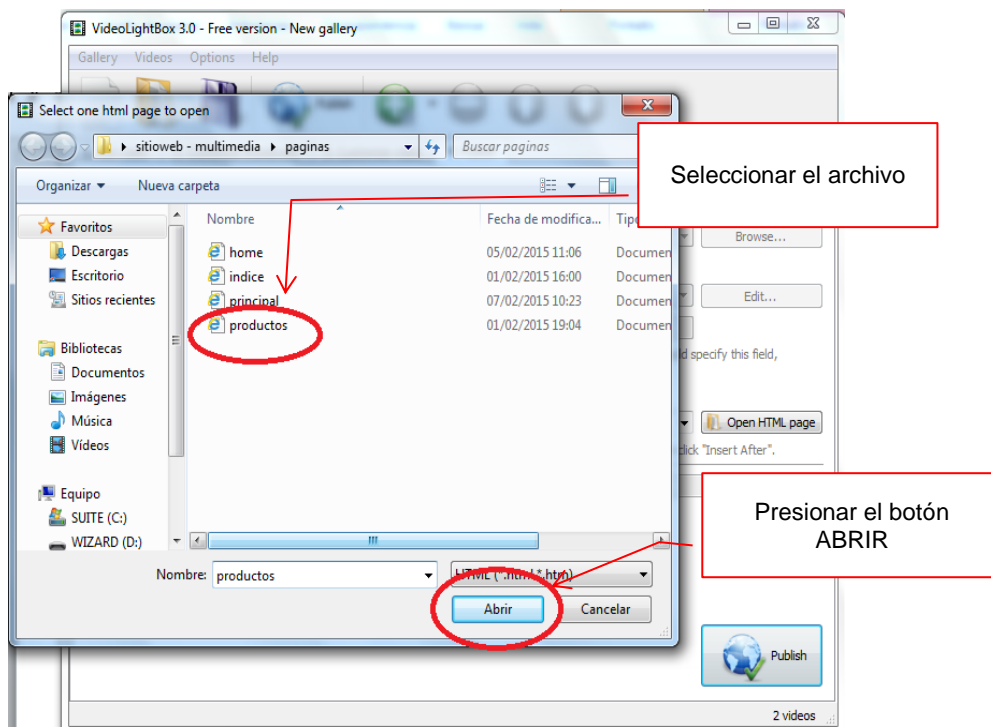
Control de la calidad de las imágenes en miniatura de la producción mediante la definición de "calidad en miniatura" parametro (0% -100%).  
Usted puede establecer los distintos tamaños de las miniaturas y resolución de video.



**Paso 3 - Añadir Vídeo LightBox dentro de su propia pagina.** Video LightBox genera un código especial. Se puede pegar en cualquier lugar de la página donde desee agregar su video.

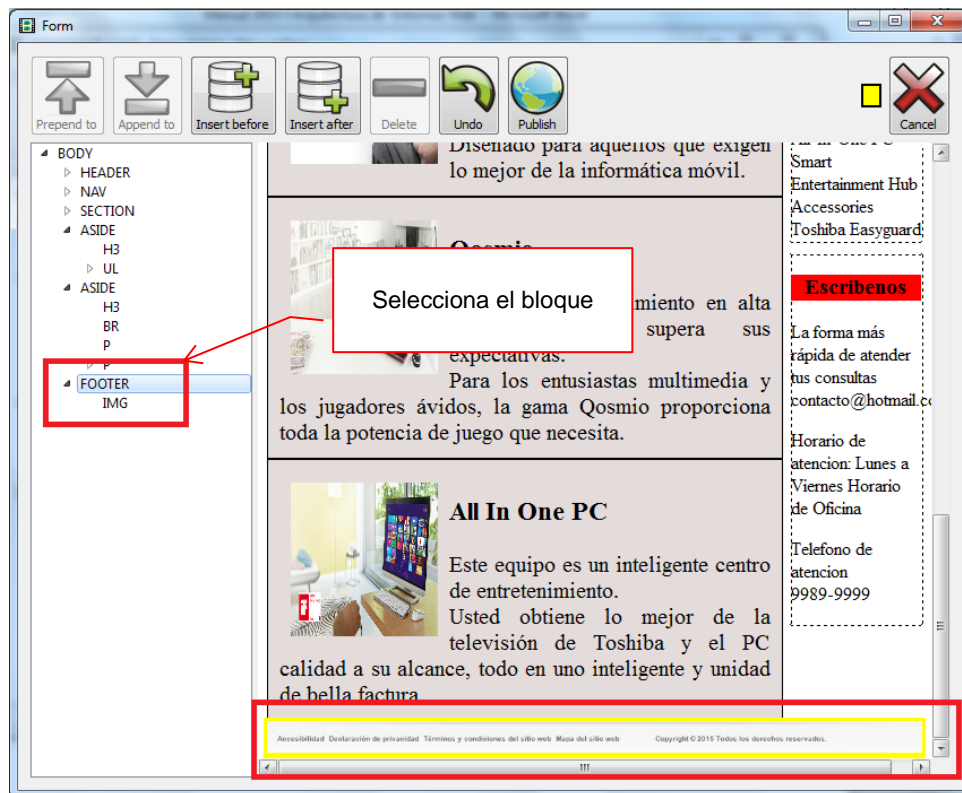


Abrir el archivo html el cual se agregará los videos. A continuación, presiona el botón PUBLISH

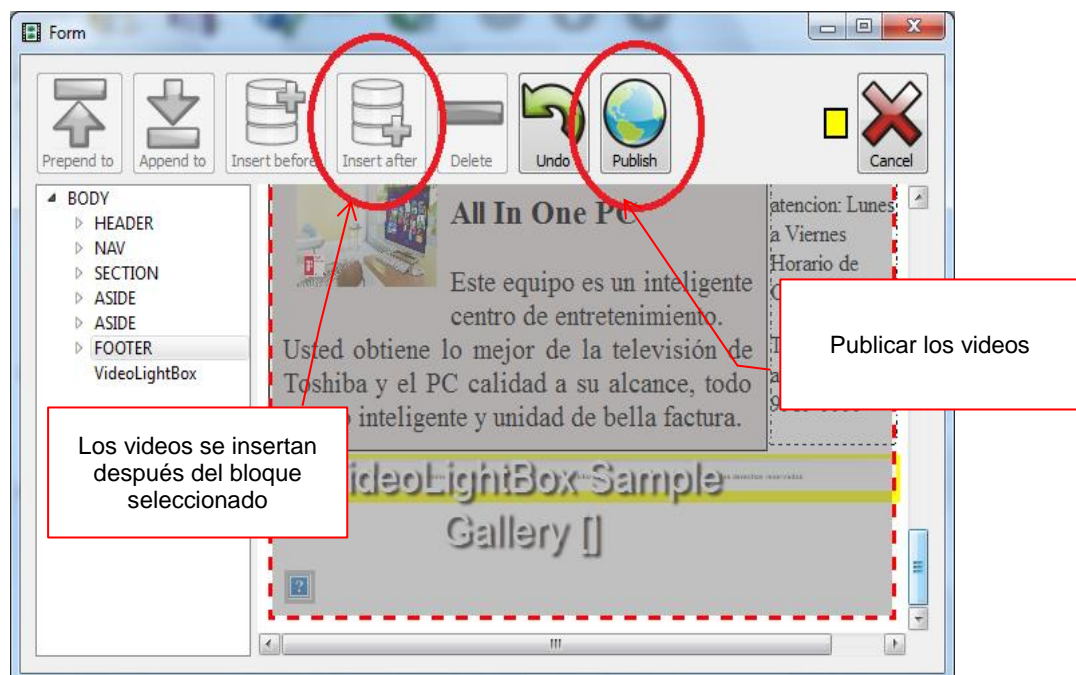




Selecciona el bloque donde se agregarán los videos: footer. A continuación, presiona el botón InsertBefore o InsertAfter.



Al presionar InsertAfter aparecerá la ubicación donde se agregarán los videos, luego presiona el botón Publish para publicar los videos en la página



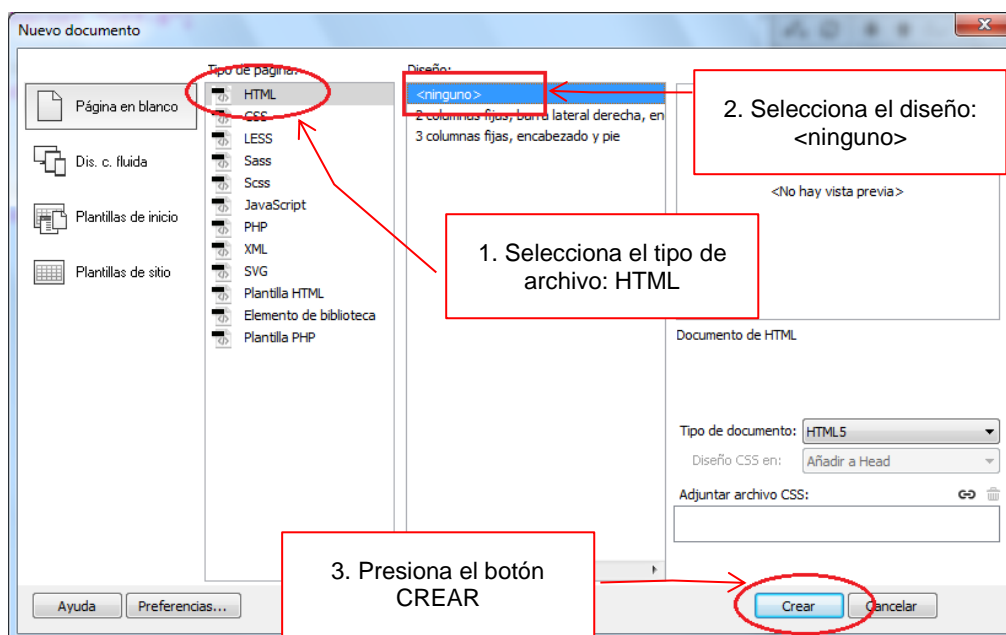
## LABORATORIO 7.1

Se nos pide diseñar una página HTML para la empresa Toshiba, donde agregue videos en la barra lateral de la página, tal como se muestra



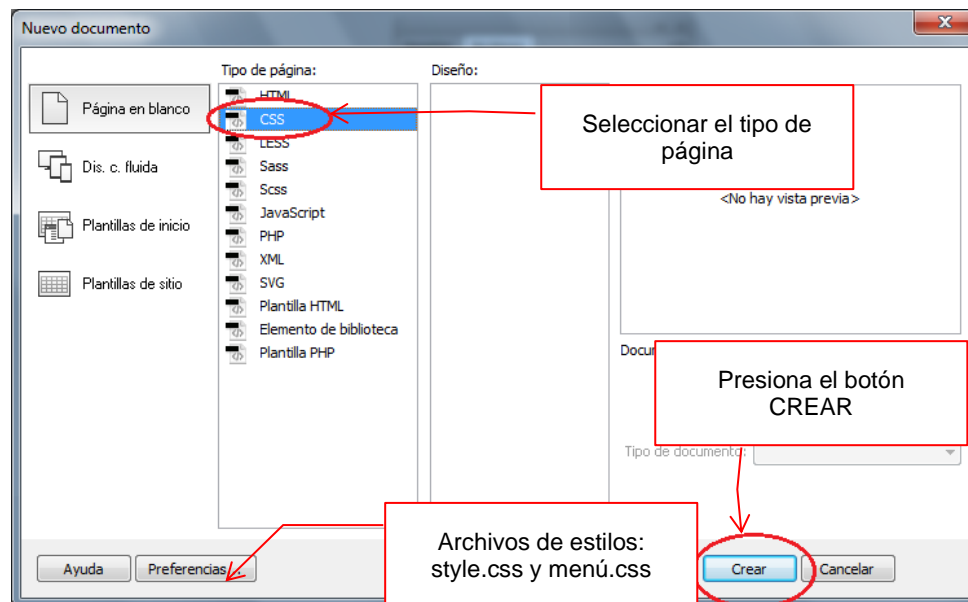
## Agregando la pagina HTML

A continuación, creamos un archivo HTML, el cual no tendrá diseño (ninguno), tal como se muestra. Guardar la página con el nombre de principal.html, en la carpeta páginas.

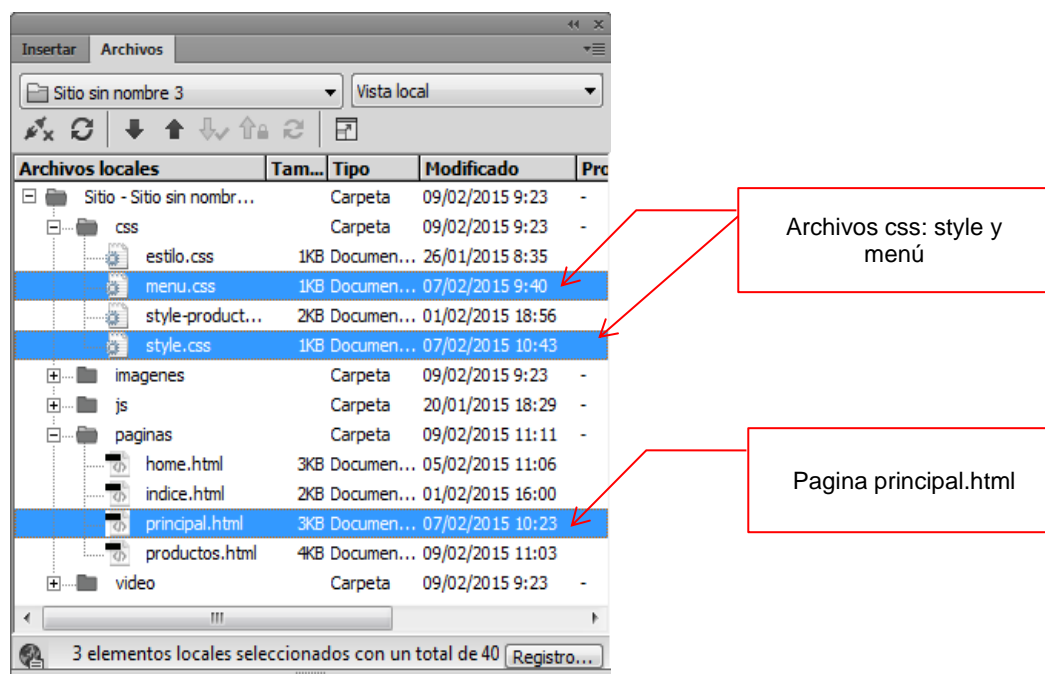


## Agregando un archivo CSS

Antes de iniciar con el diseño de la página HTML, agregamos dos archivos CSS, para definir las reglas de estilos que utilizará la página. Los archivos se llamarán: style.css y menú.css. Creado cada archivo, procedemos a guardarlo en la carpeta css.

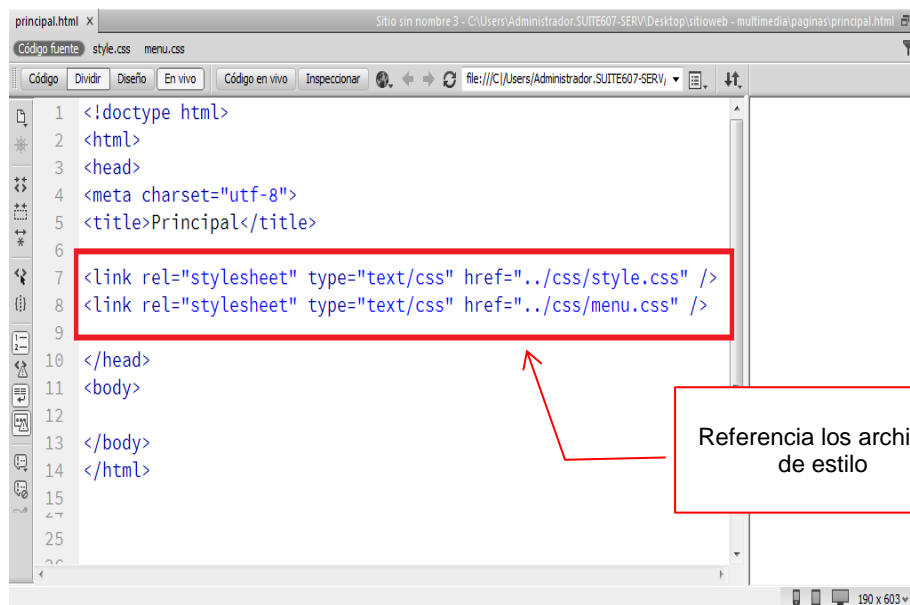


Al guardar los archivos estos se visualizan en la carpeta del Sitio Web de esta manera.



## Enlazando las hojas de estilo a la pagina HTML

En la página home, enlazamos los archivos de estilo a través de la etiqueta <link> el cual hace una referencia a los archivos style.css y menú.css, tal como se muestra

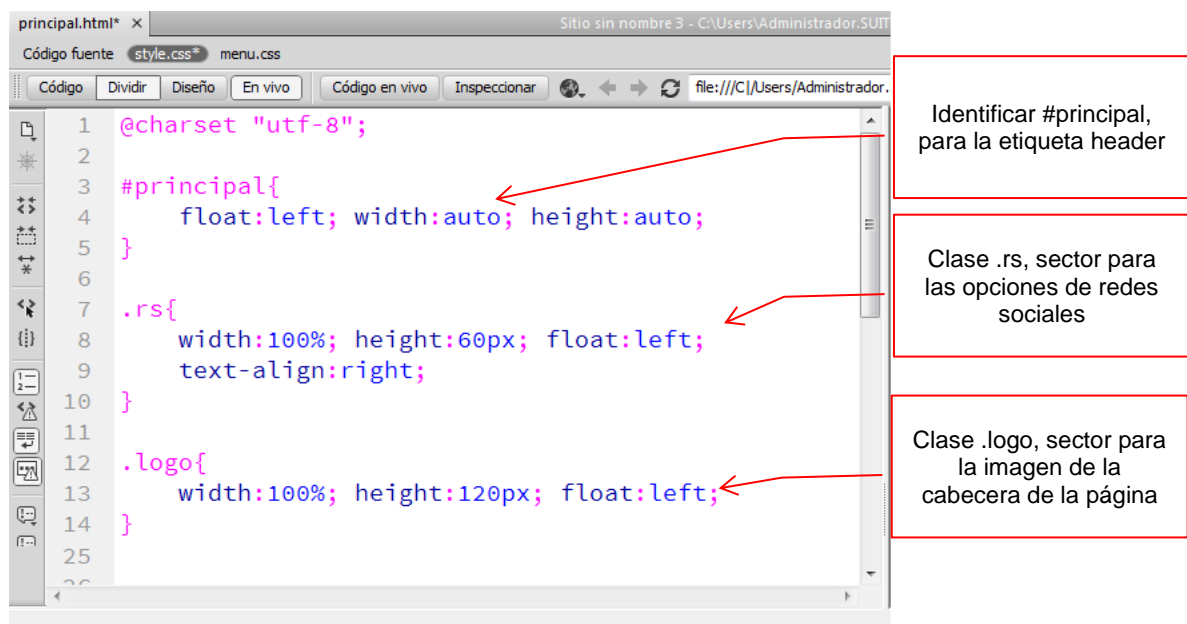


## Agregando reglas de estilo a la cabecera de la página

En el archivo style.css, definas las reglas de estilo para la cabecera:

- #principal, identificador que define la reglas para la etiqueta <header>
- .rs, clase que define el sector para las opciones de redes sociales
- .logo, clase que define el sector para la imagen de la página

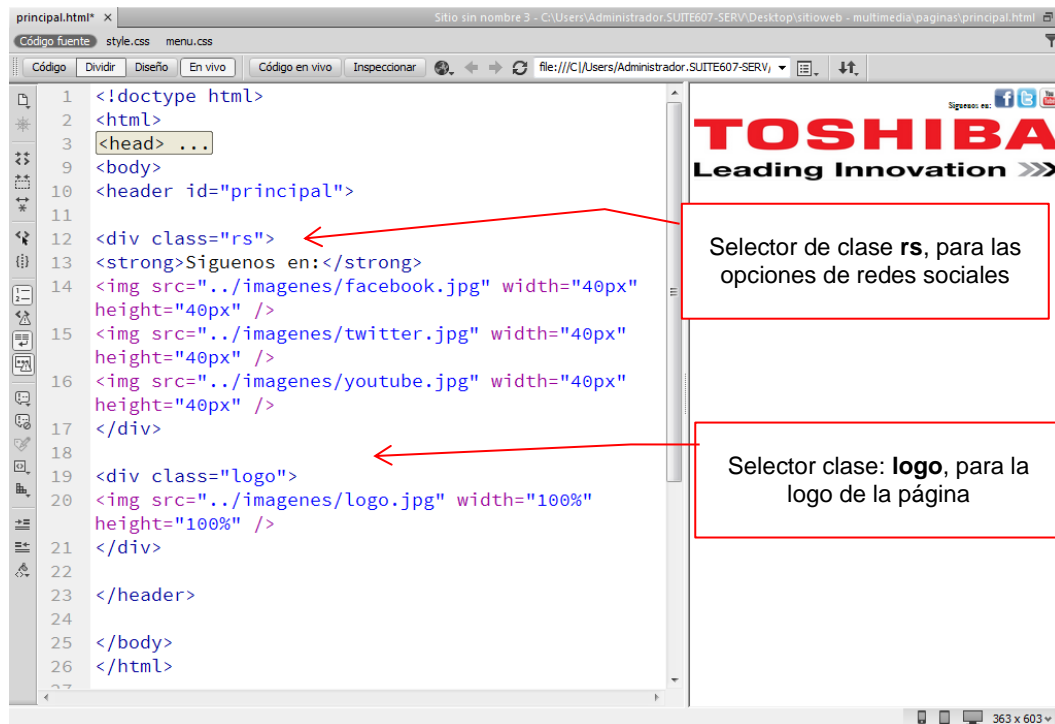
A continuación, guardar el archivo estilo.css: CTRL+S



## Agregando la etiqueta <header> y su contenido en la pagina HTML

Definida las reglas en style.css, dentro del body defina la etiqueta <header> asignando el identificador **principal**. Agrega dos bloques div:

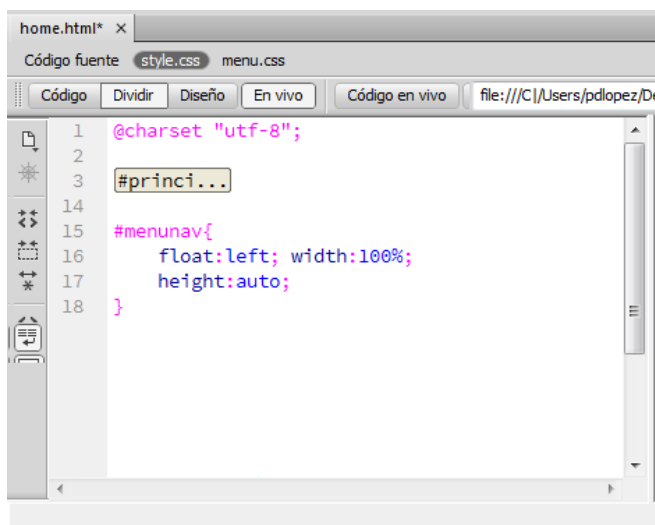
- <div class="rs"> sector para las opciones de redes sociales
- <div class="logo"> sector para el logo de la pagina.



## Agregando reglas de estilo al menú de navegación <nav>

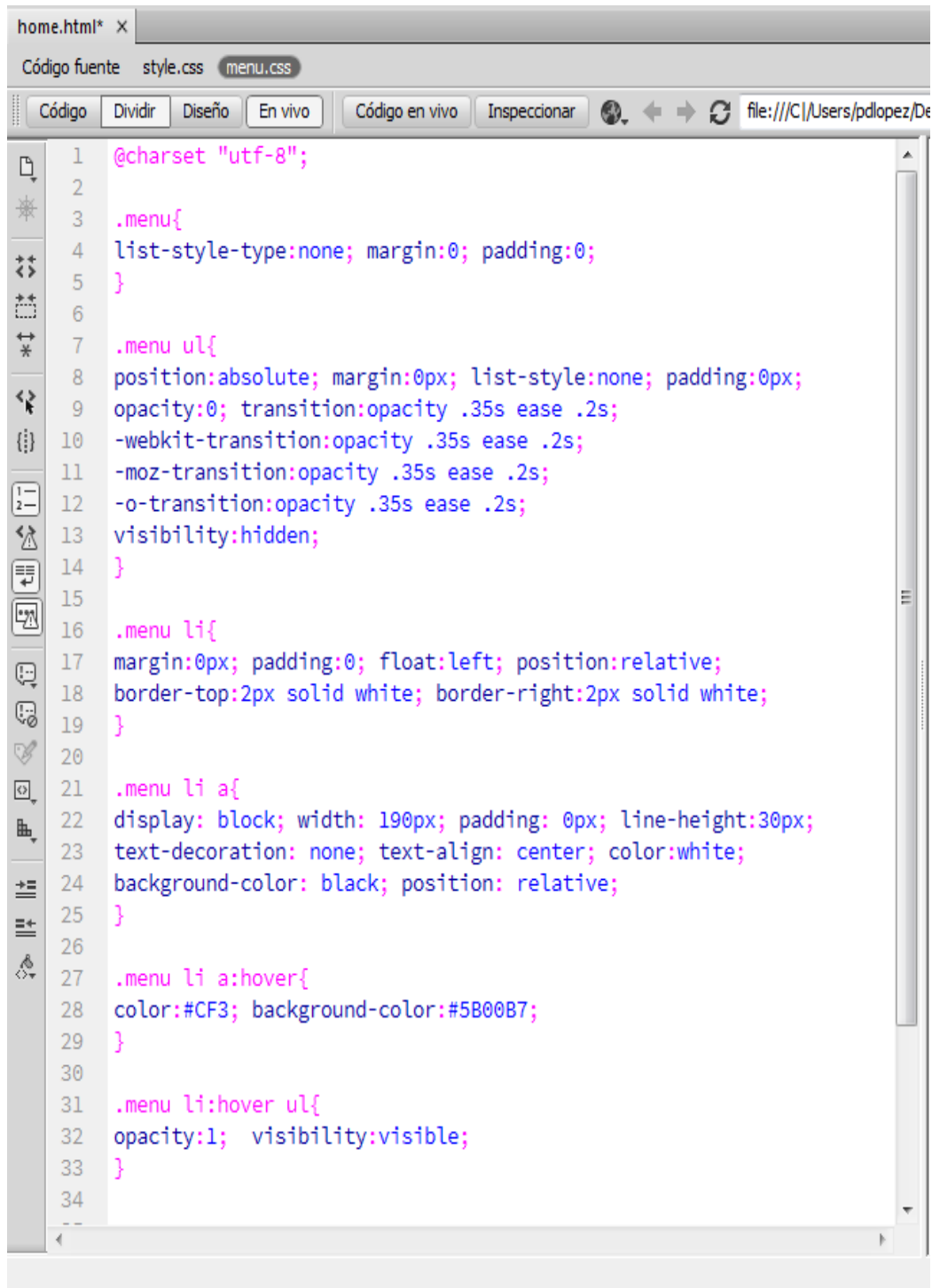
En el archivo style.css, definas las reglas de estilo para el menú de navegacion:

- #menunav, identificador que define las reglas para el bloque <nav>



## Definiendo las reglas para el archivo menú.css

En el archivo menú.css defina las reglas para crear un menú desplegable.



The image shows a web browser window with the developer tools open to the 'menu.css' file. The code defines the following CSS rules:

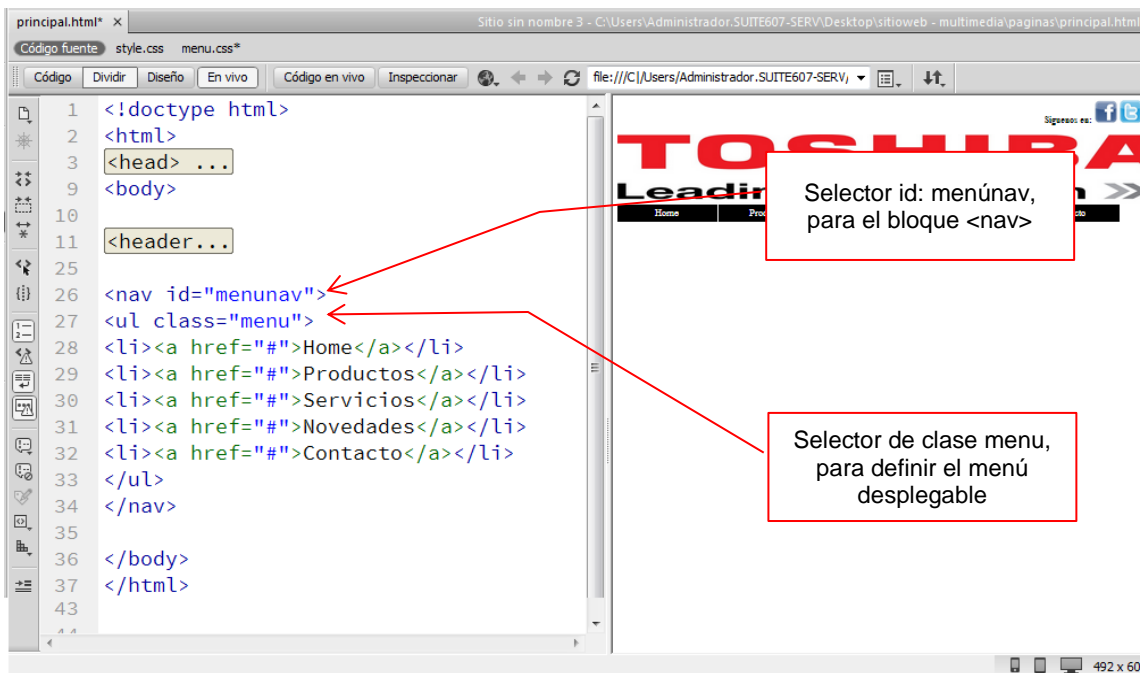
```

1  @charset "utf-8";
2
3  .menu{
4  list-style-type:none; margin:0; padding:0;
5  }
6
7  .menu ul{
8  position:absolute; margin:0px; list-style:none; padding:0px;
9  opacity:0; transition:opacity .35s ease .2s;
10 -webkit-transition:opacity .35s ease .2s;
11 -moz-transition:opacity .35s ease .2s;
12 -o-transition:opacity .35s ease .2s;
13 visibility:hidden;
14 }
15
16 .menu li{
17 margin:0px; padding:0; float:left; position:relative;
18 border-top:2px solid white; border-right:2px solid white;
19 }
20
21 .menu li a{
22 display: block; width: 190px; padding: 0px; line-height:30px;
23 text-decoration: none; text-align: center; color:white;
24 background-color: black; position: relative;
25 }
26
27 .menu li a:hover{
28 color:#CF3; background-color:#5B00B7;
29 }
30
31 .menu li:hover ul{
32 opacity:1; visibility:visible;
33 }
34

```

## Agregando la etiqueta <nav> y sus enlaces en la página HTML

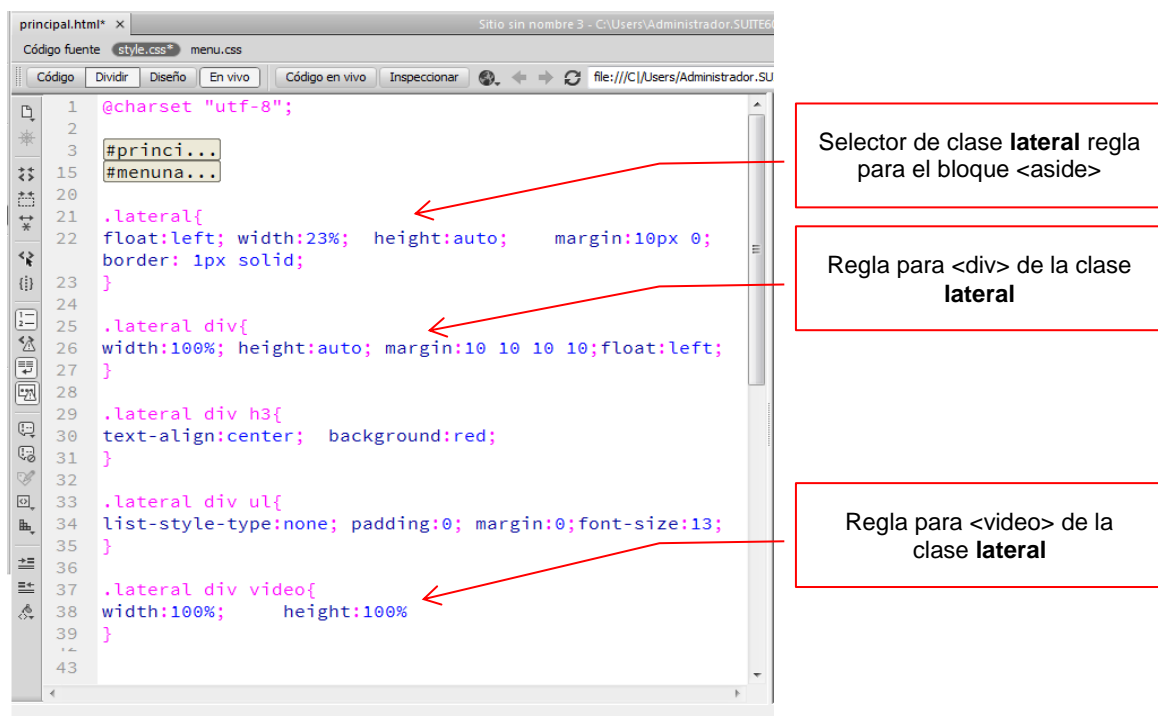
Definida las reglas, defina la etiqueta <nav> agregando una lista no ordenada, sus ítems y enlaces <a href>. Presiona la tecla F5 para visualizar el diseño de la página.



## Agregando las reglas en la barra lateral <aside>

En el archivo CSS, style-productos.css defina las reglas para barra lateral.

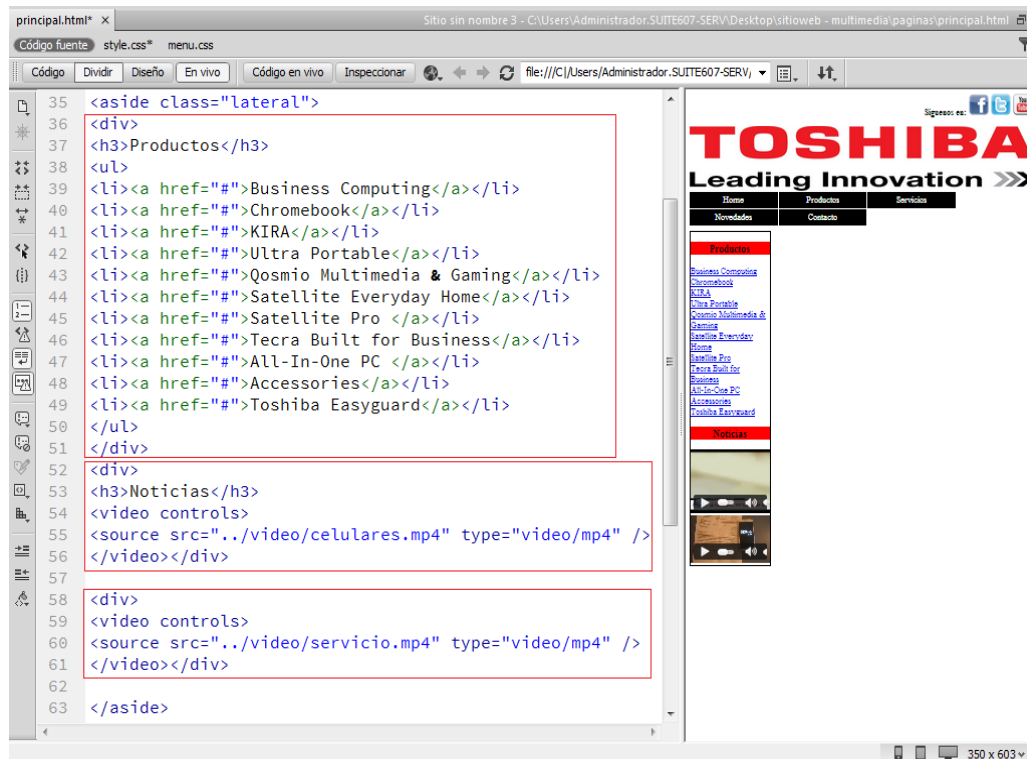
- .lateral, regla para el bloque <aside> donde su posición flotante es derecha
- .lateral div, regla para el bloque div del aside
- .lateral div h3, regla para la etiqueta <h3> del bloque <aside>
- .lateral div ul, regla para la lista <ul> dentro de cada bloque article





## Agregando la etiqueta <aside> para la página HTML

Definida las reglas en el archivo css, defina la etiqueta <aside> para mostrar información adicional de los productos y los videos. Presiona la tecla F5 para visualizar el diseño.

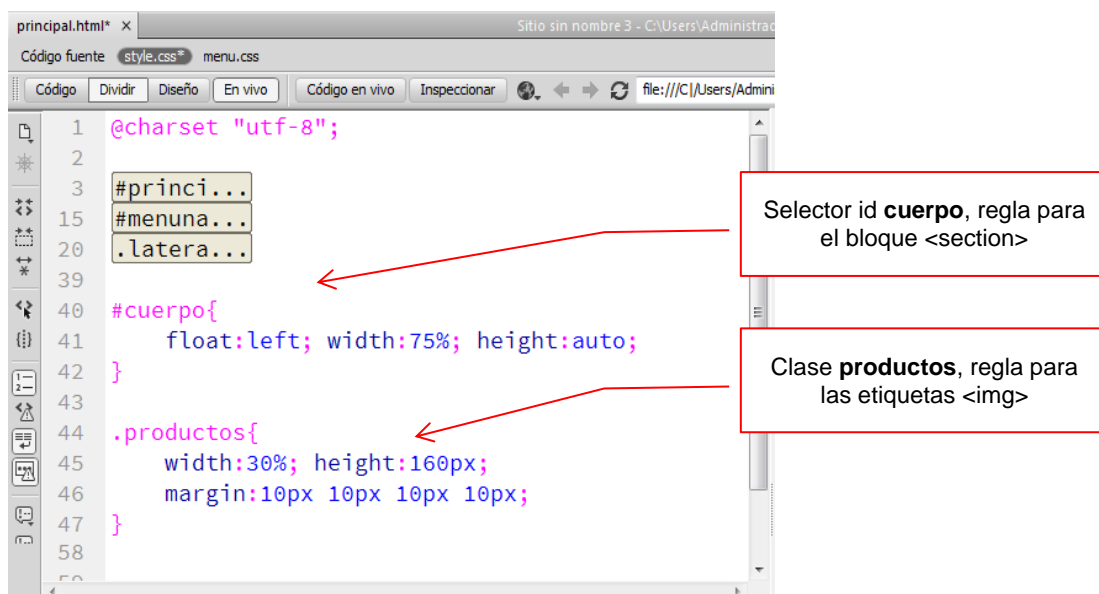


## Agregando las reglas en el cuerpo de la pagina <section>

En el archivo CSS, style-productos.css defina las reglas para el cuerpo de la página

- #cuerpo, regla para el bloque <section>, su ancho será 75% de la página
- .productos, regla para las imagenes

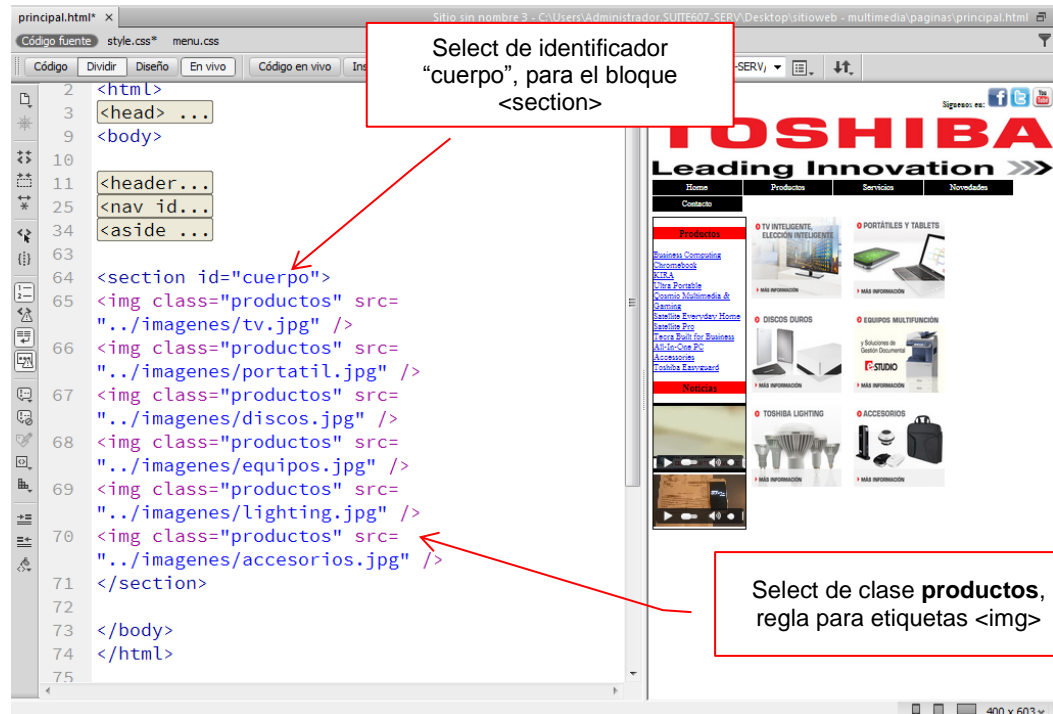
A continuación. guardar el archivo estilo.css: CTRL+S





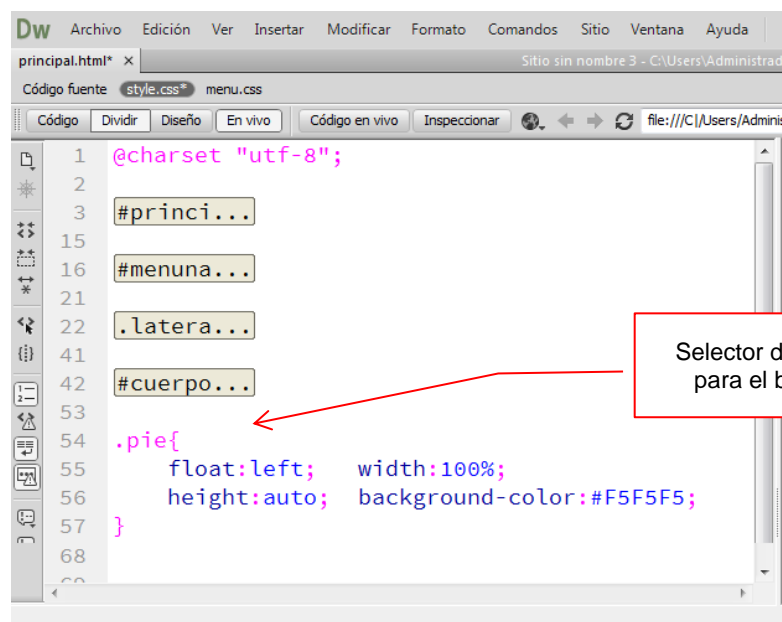
## Agregando la etiqueta <section> para la página HTML

Definida las reglas, defina la etiqueta <section> para promocionar los productos. Presiona la tecla F5 para visualizar el diseño.



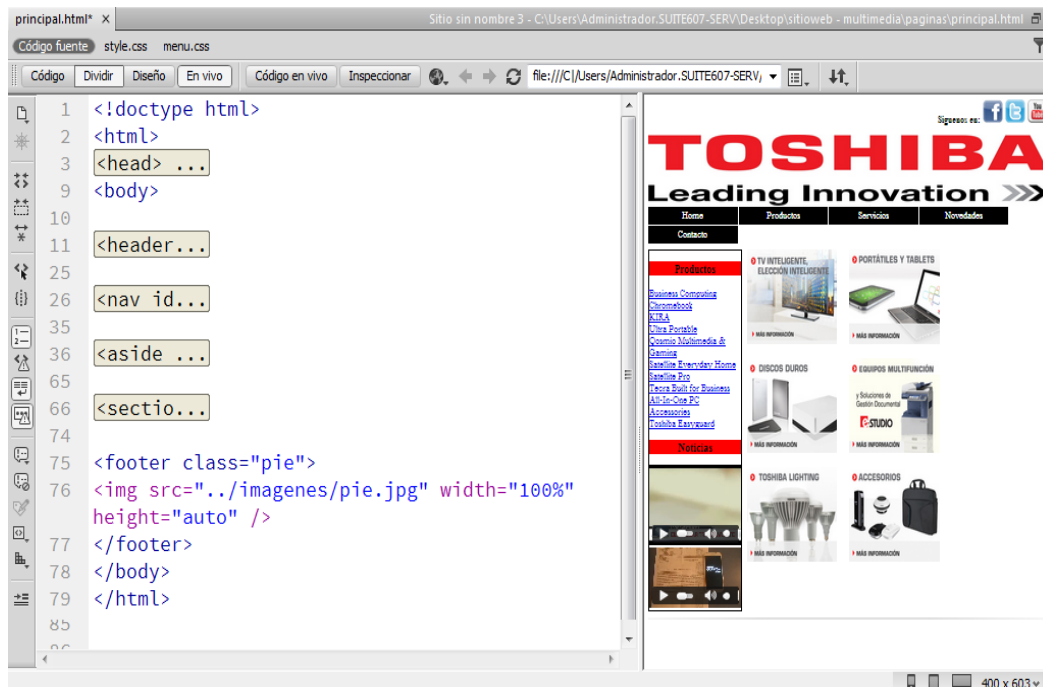
## Agregando el pie de página <footer>

Definida las reglas en style.css, defina la etiqueta <footer> agregando su identificador pie, tal como se muestra. Presiona la tecla F5 para visualizar el diseño.

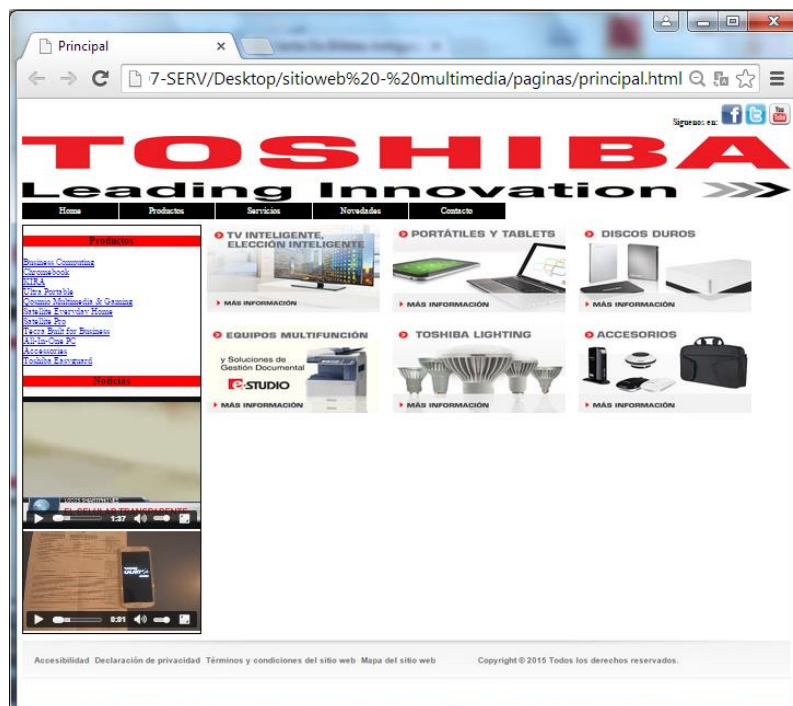


## Agregando el pie de página <footer>

Definida las reglas en style.css, defina la etiqueta <footer> agregando su identificador **pie**, tal como se muestra. Presiona la tecla F5 para visualizar el diseño.



Para ejecutar a través del navegador, primero guardar la página: CTRL+S y presiona la tecla F12 para ejecutar la página con el navegador.



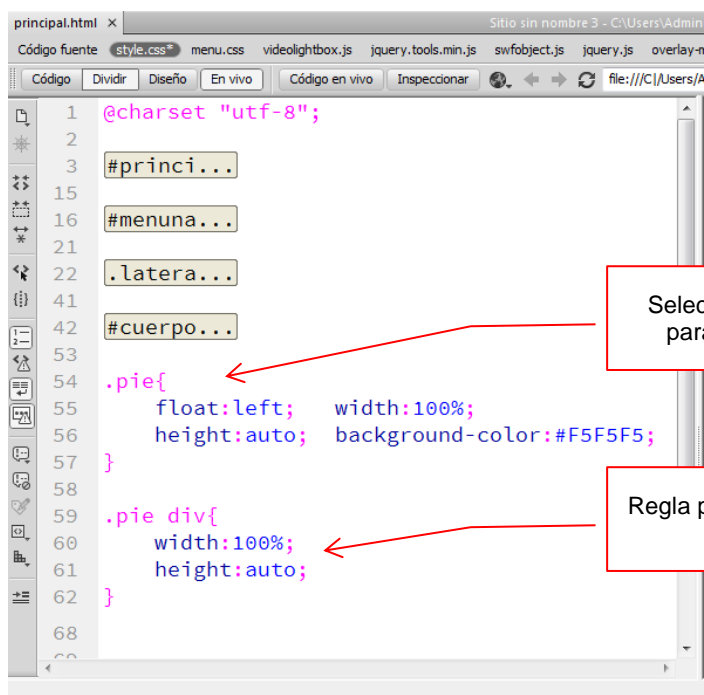
## LABORATORIO 7.2

Se nos pide diseñar una página HTML para la empresa Toshiba, donde, utilizando la herramienta VideoLightBox, agregue videos en el bloque footer.



### Agregando el pie de página <footer>

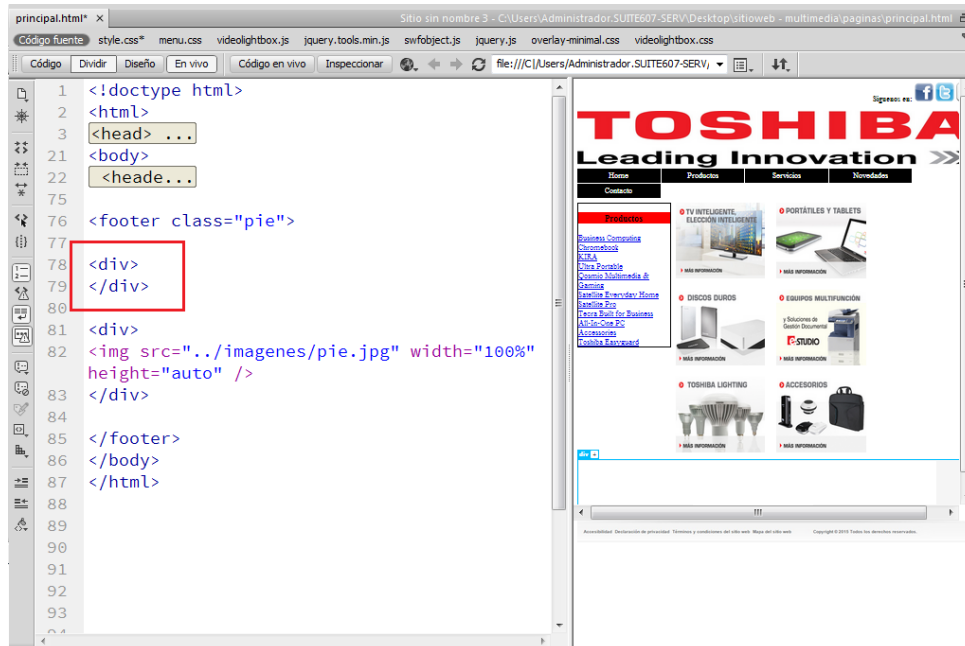
Definida las reglas en style.css, defina la etiqueta <footer> agregando su identificador **pie**, tal como se muestra. Presiona la tecla F5 para visualizar el diseño.



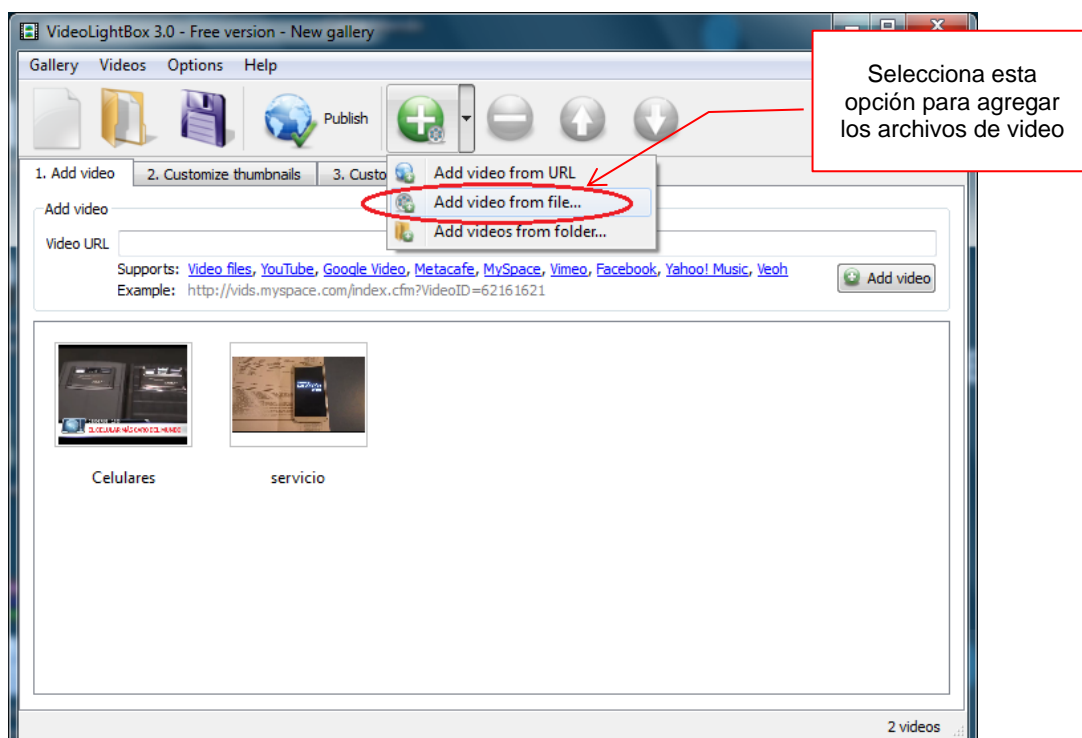
Selector de clase **pie** regla para el bloque <footer>

Regla para el bloque <div> del sector <footer>

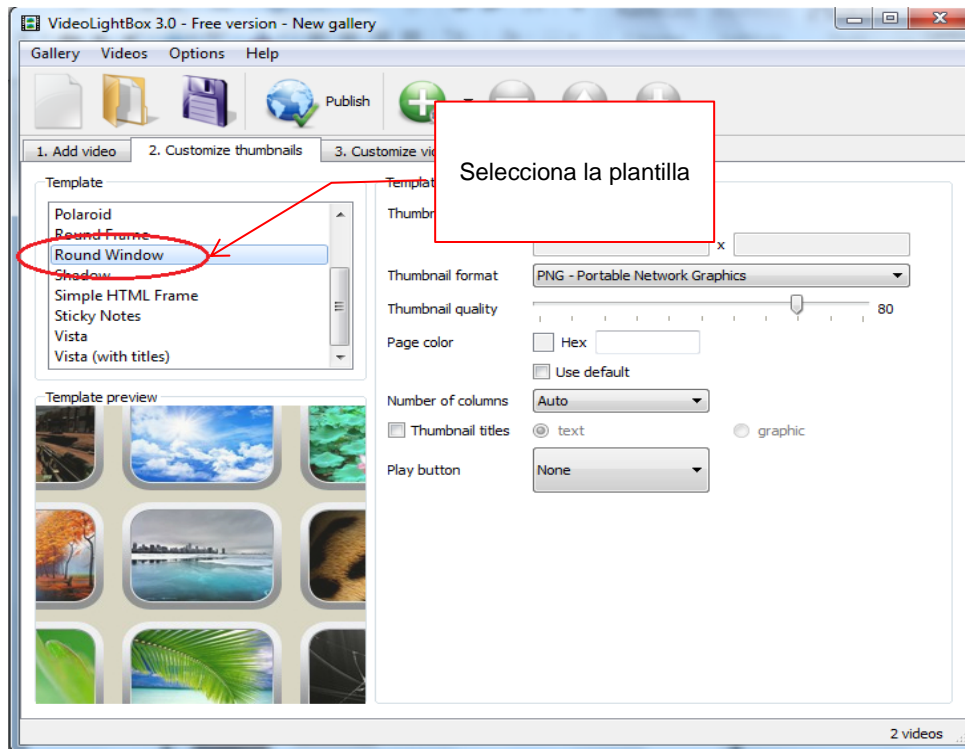
En la página agrega la estructura <div> tal como se muestra.



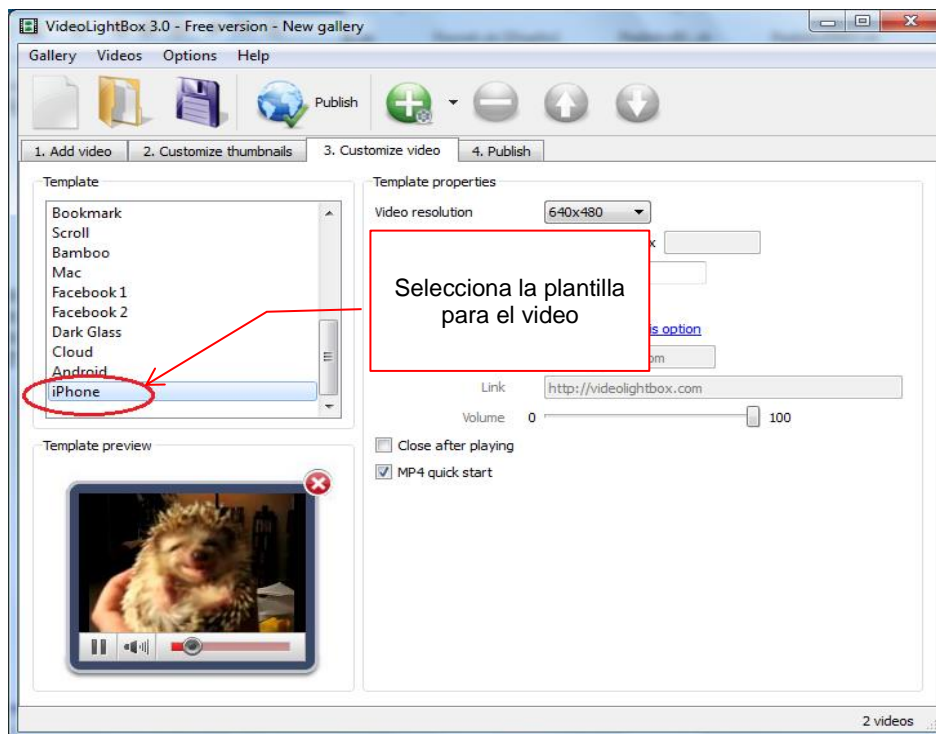
## Agregar los videos al VideoLight



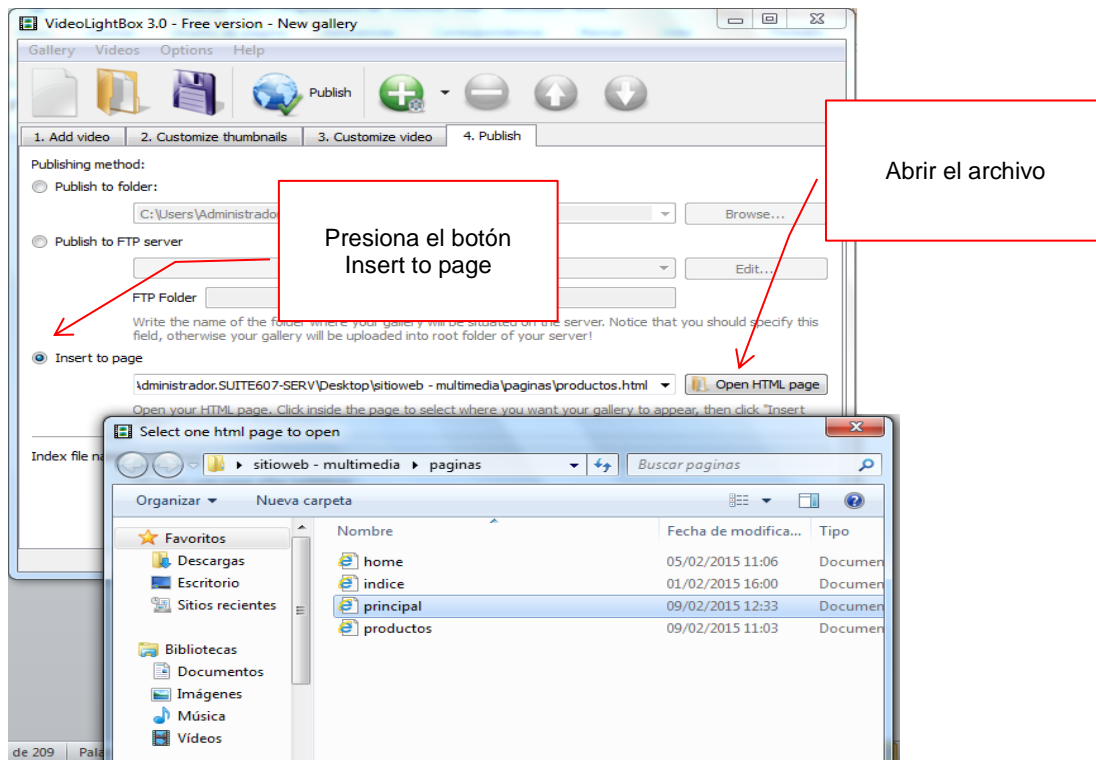
Selecciona la plantilla de la galería, tal como se muestra.



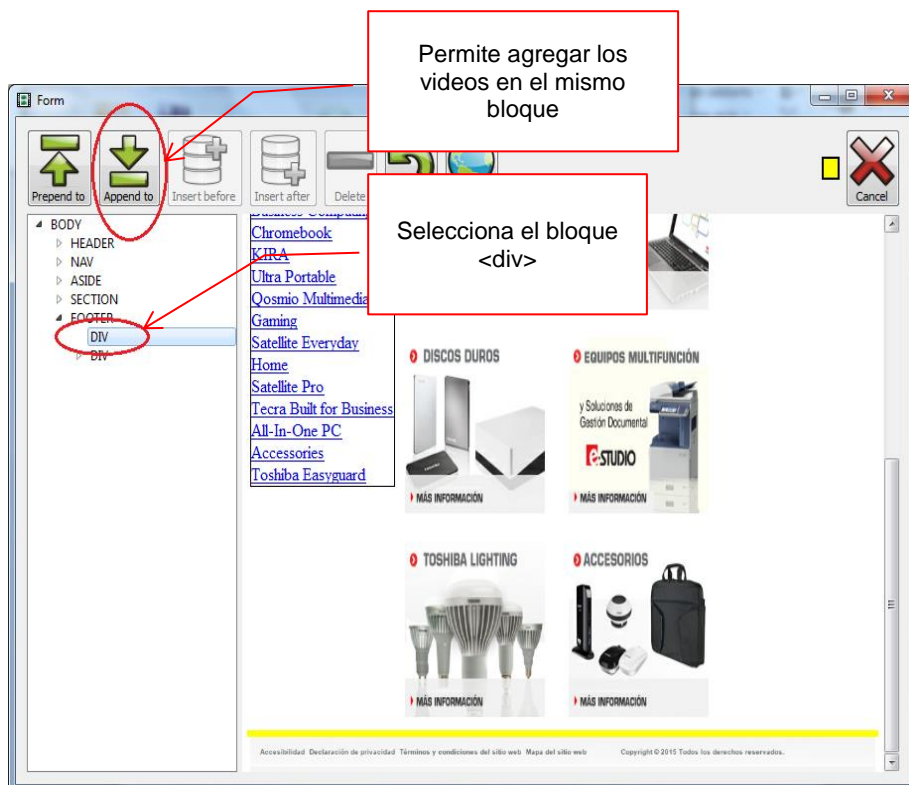
Selecciona la plantilla del video, tal como se muestra.



Para publicar, selecciona la opción Insert to Page, y buscar el archivo para abrir la pagina HTML. Presiona el botón Publish.

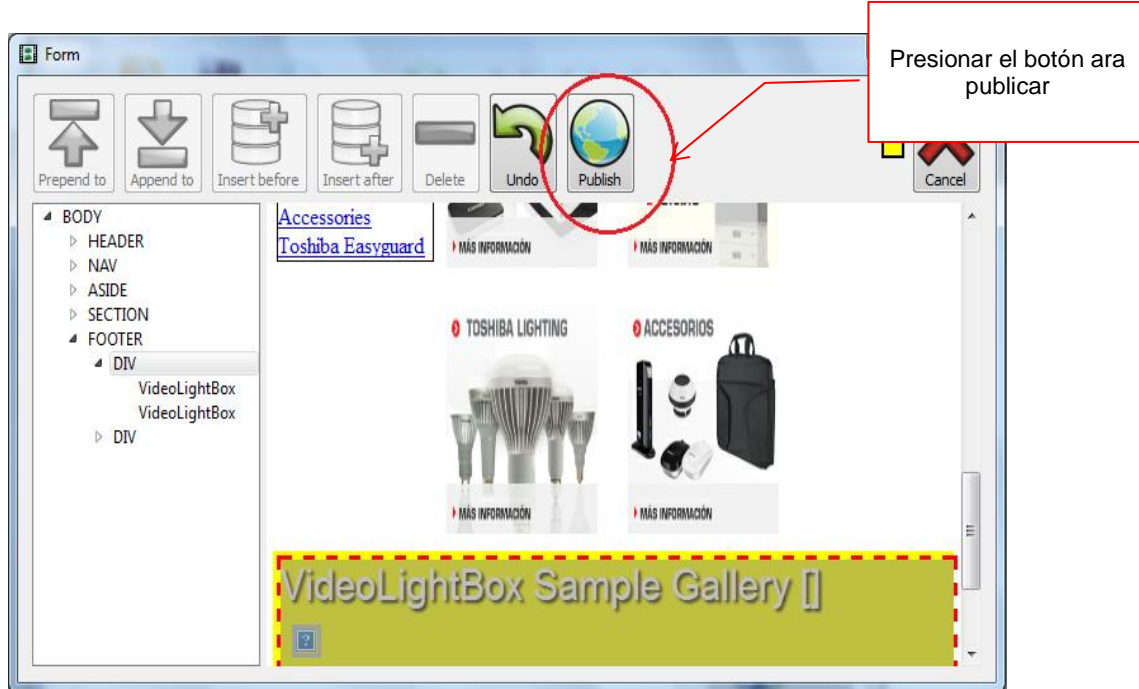


A continuación, presiona el botón PUBLISH. Selecciona el bloque <div> donde se publicarán los videos, tal como se muestra y presiona el botón Append to, tal como se muestra.





Al presionar Append to, se muestra la región donde se agrega la galería, presionar el botón Publish.



Para visualizar el resultado del VidioLightBox, visualizar la pagina por el navegador, el cual muestra los videos a continuación.



## Resumen

- 📖 Algunos de los cambios más notables en HTML5 es la adición de etiquetas para soportar elementos multimedia. Esto incluye la etiqueta **<audio>** para reproducir sonidos, **<video>** para videos, todo esto dentro del web browser y sin depender de software externo.
- 📖 El elemento **<audio>** trabaja del mismo modo y comparte varios atributos con el elemento **<video>**: **src**, especifica la URL del archivo a ser reproducido; **controls**, activa la interface que cada navegador provee por defecto para controlar la reproducción del audio; **autoplay**, reproducción automática; **loop**, el navegador reproducirá el audio una y otra vez de forma automática; **preload**, Este atributo puede tomar tres valores diferentes: none, metadata o auto.
- 📖 Entre los atributos de la etiqueta **<video>**: **poster**, indica la imagen que el navegador debe mostrar mientras el vídeo se está descargando, o hasta que el usuario reproduce el vídeo; **height**, **width**: indican al navegador el tamaño del vídeo en pixels. Si no se indican estas medidas, el navegador utiliza las medidas definidas en el vídeo de origen, si están disponibles.
- 📖 Una vez que disponemos del vídeo o el audio en los distintos formatos, es necesario indicar todas las ubicaciones de estos formatos, para que sea el navegador el que decida que formato reproducir. Para especificarlos tendremos que hacerlo de manera separada utilizando el elemento **<source>**.
- 📖 El elemento HTML inline frame (**<iframe>**) permite incorporar otra página HTML en la página actual. Cada contexto de navegación tiene su propia historia, sesión y documento activo. El contexto de navegación que incluye el contenido implícito se llama contexto de navegación principal. El contexto de navegación de nivel superior (que no tiene padre) es típicamente la ventana del navegador
- 📖 Para incrustar un video de youtube en un documento HTML
 

```
<iframe width="560" height="315" src="//www.youtube.com/embed/5AKY58ZSJGs" frameborder="0" allowfullscreen="">
</iframe>
```
- 📖 Video Lightbox JS es un programa gratuito que te ayuda a integrar fácilmente vídeo al sitio web, página web o blog, en unos pocos clics sin tener que escribir una sola línea de código
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 <https://developer.mozilla.org/es/docs/Web/HTML/UsandoaudioyvideoconHTML5>

🔗 <http://www.arkaitzgarro.com/html5/capitulo-6.html>

🔗 <http://ocw.uv.es/ingenieria-y-arquitectura/programacionmultimedia/t2-3-html5.pdf>

🔗 <http://www.um.es/docencia/barzana/DAWEB/Desarrollo-de-aplicaciones-web-teoria-html5-multimedia.html>

🔗 <https://developer.mozilla.org/es/docs/Web/HTML/Elemento/iframe>

🔗 <http://personales.upv.es/moimacar/download/html5.pdf>





# TRANSICIÓN Y ANIMACIÓN CON CSS

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3 y animaciones.

## TEMARIO

### 5.1 Tema 8: Transición y Animación

#### 5.1.1 Transiciones

##### 5.1.1.1 Propiedades de las transiciones

##### 5.1.1.2 Ejemplos

#### 5.1.2 Animaciones

##### 5.1.2.1 KeyFrames

##### 5.1.2.2 Propiedades de las animaciones

##### 5.1.2.3 Ejemplos

## ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5, hojas de estilo CSS3, bloques de contenidos con transiciones y animaciones
- Los alumnos diseñan un menú animado en una página web



## 5.1 TRANSICIÓN Y ANIMACIÓN

Las nuevas etiquetas HTML5 y CSS3 proporcionan a los diseñadores web cada vez más recursos para el desarrollo de sus sitios web.

CSS3 pretende estandarizar dos nuevas características: las transiciones y las animaciones que actúan sobre el estado de un elemento transformándolo de manera gradual. Con esto se consiguen efectos muy agradables que mejoran notablemente la experiencia del usuario al interactuar con la página.

Estas nuevas características son muy modernas por lo que los navegadores las van incluyendo poco a poco en sus nuevas versiones. De hecho, Internet Explorer todavía no las soporta en su versión 9, aunque Microsoft asegura que las incluirá a partir de la versión 10.

En este capítulo vamos a explicar qué son las transiciones y animaciones en CSS; mencionaremos sus características y aprenderemos a transformar y animar elementos de nuestras páginas web únicamente con CSS. Un hecho que antes era impensable sin ayuda de Flash o Javascript.

### 5.1.1 Transición

La transición es el efecto de un cambio en el estado de un elemento de forma gradual en un intervalo de tiempo. Cuando hablamos de cambio de estado nos referimos a cambios en los valores de sus propiedades.

Las transiciones únicamente son soportadas por los navegadores: Firefox, Chrome, Safari y Opera. Internet Explorer (al menos hasta su versión 9) no las soporta.

#### 5.1.1.1 Propiedades de la Transición

Para aplicar una transición sobre un determinado elemento, contamos con las siguientes propiedades:

- **transition-property:** Determina la(s) propiedad(es) a ser animadas; puede ser cualquier propiedad CSS. El valor por defecto es "all" que la transición se aplicara sobre todas las propiedades del elemento. Un valor valido para esta propiedad seria, por ejemplo "width"
- **transition-duration:** Indica la duración de la animación del inicio al fin en segundos. Por defecto el valor es 0, ejemplos de valores validos para esta propiedad pueden ser: 2s (2 segundos), 300ms (300 milisegundos), etc.
- **transition-timing-function:** Especifica la velocidad que se desarrollara la transición utilizando las llamadas curvas de Bezier. Asi podemos programar la transición para que empiece a una velocidad, luego frene y después se acelere o darle movimiento. Los valores que acepta esta propiedad son los siguientes:
  - **linear:** se produce en forma de progresión lineal. Tarda lo mismo desde que empieza hasta que acaba.
  - **ease:** la transición se produce en forma lenta al principio, luego mas rápido, y termina otra vez lenta (valor por defecto).
  - **ease-in:** la transición comienza lenta y luego va mas rápido
  - **ease-out:** la transición comienza mas rápido y termina lenta
  - **ease-in-out:** la transición comienza y acaba lenta. Igual que 'ease' pero de forma más notable.

- **cubic-bezier (n, n, n, n):** permite definir nuestro propio intervalo de progresión. n: valor entre 0 y 1.
- **transition-delay:** Tiempo en que la animación debe pausarse antes de comenzar. Por defecto es 0.
- **Transition:** permite unificar todas las propiedades anteriores en una sola.

La lista de propiedades que se pueden modificar mediante transiciones:

- background-color (color)
- background-image (solo gradientes)
- background-position (porcentaje y longitud)
- border-bottom-color (color)
- border-bottom-width (longitud)
- border-color (color)
- border-left-color (color)
- border-left-width (longitud)
- border-right-color (color)
- border-right-width (longitud)
- border-spacing (longitud)
- border-top-color (color)
- border-top-width (longitud)
- border-width (longitud)
- bottom (longitud y porcentaje)
- color (color)
- crop (rectángulo)
- font-size (longitud y porcentaje)
- font-weight (número)
- grid-\* (diversos valores)
- height (longitud y porcentaje)
- left (longitud y porcentaje)
- letter-spacing (longitud)
- line-height (número, longitud y porcentaje)
- margin-bottom (longitud)
- margin-left (longitud)
- margin-right (longitud)
- margin-top (longitud)
- max-height (longitud y porcentaje)
- max-width (longitud y porcentaje)
- min-height (longitud y porcentaje)
- min-width (longitud y porcentaje)
- opacity (número)
- outline-color (color)
- outline-offset (entero)
- outline-width (longitud)
- padding-bottom (longitud)
- padding-left (longitud)
- padding-right (longitud)
- padding-top (longitud)
- right (longitud y porcentaje)
- text-indent (longitud y porcentaje)
- text-shadow (sombra)
- top (longitud y porcentaje)
- vertical-align (palabras clave, longitud y porcentaje)

- visibility (visibilidad)
- width (longitud y porcentaje)
- word-spacing (longitud y porcentaje)
- z-index (entero)
- zoom (número)

### 5.1.1.2 Ejemplos

En este ejemplo de transiciones vamos a aplicar un zoom en un texto. Cuando un usuario pase el ratón por encima de un texto contenido en un párrafo, el tamaño de letra se incremente y cambie el color de la letra:

- La transición se aplique a la propiedad font-size (transition-property: font-size).
- La transición dure 300 milisegundos (transition-duration: 300s).
- La transición se produzca de forma siempre a la misma velocidad (transition-timing-function: linear).
- La transición se produzca sin retardo (transition-delay: 0s).

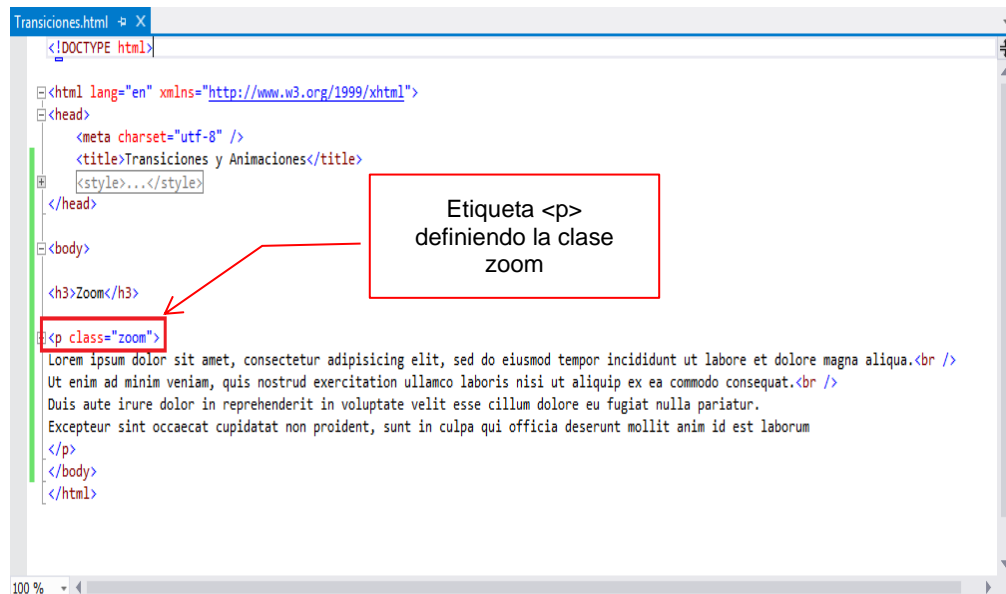
Primero definimos la clase zoom, en dicha clase definimos la transición del texto. Esta transición es aplicada para los navegadores Firefox, Chrome, Safari, tal como se muestra.

```
p.zoom {  
    font-size:14px;  
    color:blue;  
    text-align:justify;  
    width:250px;  
    height:auto;  
    /* Firefox */  
    -moz-transition: font-size 300ms linear 0s;  
    /* Chrome / Safari */  
    -webkit-transition: font-size 300ms linear 0s;  
    /* Default */  
    transition: font-size 300ms linear 0s;  
}
```

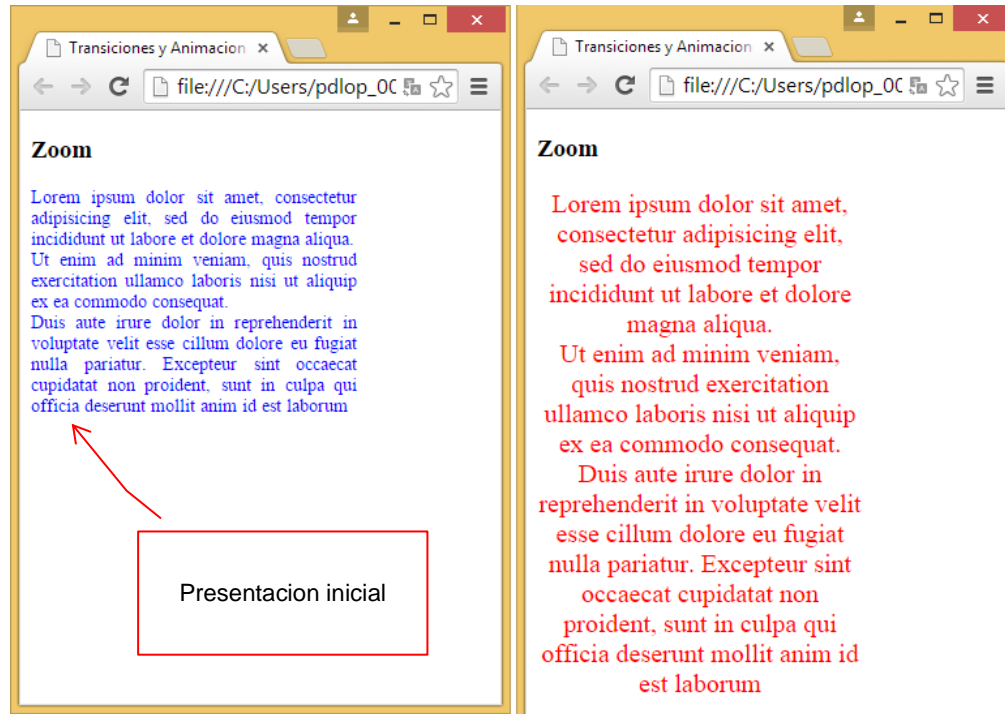
Luego defina las propiedades al evento hover de la clase zoom, donde al pasar el cursor sobre el texto aumenta el tamaño de la letra y cambia el color de la letra.

```
p.zoom:hover {  
    font-size:20px;  
    color:red;  
    text-align:center;  
}
```

En la página, primero defina la etiqueta `<p>` definiendo la clase `zoom`, para aplicar el zoom al texto redactado dentro del párrafo `<p>`.



Al finalizar el diseño de la página, ejecutarla en un navegador, Chrome. Al inicio se visualiza como la página de la izquierda; pero al colocar el cursor sobre el texto, se muestra como la página de la derecha.



El siguiente ejemplo de transición es aplicado a las imágenes; lo que haremos es que se produzca una transición en las propiedades de las imágenes: width, height y border cuando el usuario pase el mouse por encima de la imagen.

Como primer paso definiremos las propiedades de la transición:

- La transición debe aplicarse a todos los elementos que cambien (transition-property: all).
- La transición dure 2 segundos (transition-duration: 1s).
- La transición se produzca de forma lenta al principio, luego rápido y acabe lenta (transition-timing-function: ease).
- La transición se produzca con un retardo de 500ms tras pasar el mouse por encima de la imagen (transition-delay: 0.5s).

```
img.growth {  
  width:150px;  
  height:130px;  
  border:1px solid;  
  /* Firefox */  
  -moz-transition-property: all;  
  -moz-transition-duration: 2s;  
  -moz-transition-timing-function: ease;  
  -moz-transition-delay: 0.5s;  
  
  /* Chrome / Safari */  
  -webkit-transition-property: all;  
  -webkit-transition-duration: 2s;  
  -webkit-transition-timing-function: ease;  
  -webkit-transition-delay: 0.5s;  
  
  /* Default */  
  transition-property: all;  
  transition-duration: 2s;  
  transition-timing-function: ease;  
  transition-delay: 0.5s;  
}
```

A continuación, definimos las propiedades del evento hover de la imagen, tal como se muestra.

```
img.growth:hover {  
  width:250px;  
  height:230px;  
  border:2px solid;  
}
```

En la página, primero defina la etiqueta <img> definiendo la clase growth, para aplicar el zoom a la imagen



```

Transiciones.html
<!DOCTYPE html>

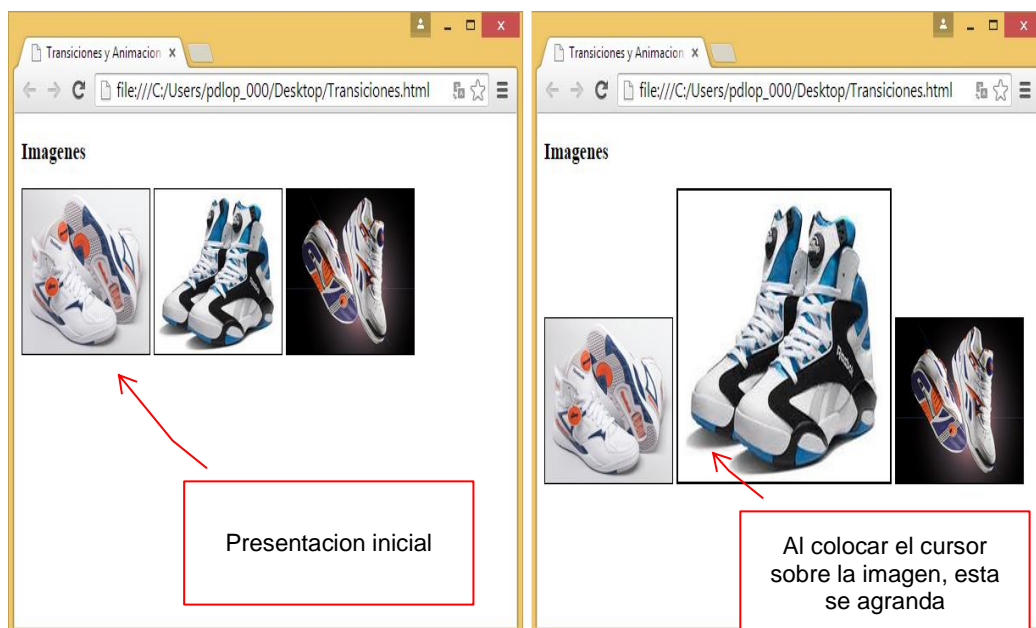
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Transiciones y Animaciones</title>
  <style>...</style>
</head>
<body>
  <h3>Imágenes</h3>

</body>
</html>
100 %

```

Al finalizar el diseño de la página, ejecutarla en un navegador, Chrome. Al inicio se visualiza como la página de la izquierda; pero al colocar el cursor sobre una imagen, se muestra como la página de la derecha.





### 5.1.2 Animaciones

Las animaciones son bastante parecidas a las transiciones. Ambas tienen en común que actúan sobre los valores de las propiedades de un elemento transformándolos gradualmente en otros valores. Sin embargo, las animaciones ofrecen bastante más control que las transiciones en el proceso de transformación del elemento.

Las transiciones se ejecutan cuando se produce un cambio en las propiedades de un elemento. Ya lo vimos en el punto anterior, cuando el usuario pasa el ratón por encima de un elemento y la CSS indica que el valor de determinadas propiedades cambia (: hover), entonces se lanza la transición.

Las animaciones son algo distintas ya que se lanzan explícitamente. No se espera a que suceda un cambio en el estado del elemento como pasaba con las transiciones. Es por ello que para usar animaciones necesitamos "keyframes". Estos "keyframes" describirán el comportamiento que tendrá la animación durante su periodo de ejecución.

#### 5.1.2.1 Propiedades de las Animaciones

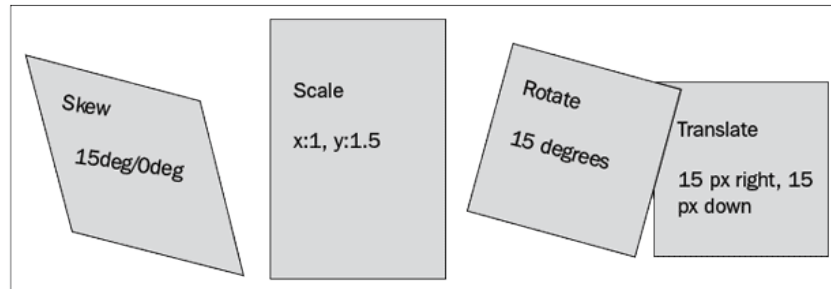
Las propiedades de las animaciones son bastante parecidas a las de las transiciones. Para poder aplicar animaciones sobre nuestros elementos contamos con las siguientes propiedades:

- **animation-name:** es el nombre de la animación que actuará sobre el elemento. Recordemos que en el punto anterior (keyframes) para definir una animación mediante un keyframe debíamos añadir también el nombre de la animación. Pues es ese nombre.
- **animation-duration:** el tiempo que tardará la animación desde su inicio hasta su fin.
- **animation-timing-function:** del mismo modo que en las transiciones, esta propiedad indica cómo progresará la animación. Acepta los mismos valores que la propiedad transition-timing-function.
- **animation-iteration-count:** indica el número de veces que se repetirá la animación. Su valor por defecto es 1. El valor "infinite" indica que se repetirá indefinidamente.
- **animation-direction:** indica el orden en el que ejecutará la animación. Acepta los valores "normal" y "alternate". Si tuviésemos dos repeticiones de la animación (animation-iteration-count: 2) y el valor de esta propiedad fuese alternate, primero se ejecutaría la animación en orden normal 0 a 100% y luego al revés 100% a 0. Si fuese normal, las dos veces se ejecutaría de 0 a 100%. El valor por defecto es normal.
- **animation-play-state:** indica si la animación está corriendo o pausada. Acepta los valores running (valor por defecto) y paused.
- **animation-delay:** igual que en las transiciones, indica el retardo con el que se iniciará la animación. Por defecto 0.
- **animation-fill-mode:** pues básicamente lo que hace esta propiedad es decidir si los valores de las propiedades que ha modificado la animación vuelven a su estado original (antes de la animación) o no. Realmente vale para ver qué se hace con los valores de las propiedades fuera del periodo de duración de la animación. Acepta los valores: "none", "forwards", "backwards" y "both". El valor por defecto es none que indica que cuando acabe la animación el elemento recuperará los valores que tenían de sus propiedades antes de iniciarse la animación.

- **animation:** permite unificar las propiedades anteriores en una sola. Sería algo así: animation: mi-animacion 3s ease 0.5s 1 normal none.

Las transformaciones de CSS3 pueden ser en 2D y en 3D, siendo estas últimas considerablemente más complejas al tener que tener en cuenta los tres ejes de coordenadas.

#### Transformaciones Básicas con CSS3



De las transformaciones de CSS3 en 2D, las más usadas son:

- **Rotate.** Rotate te permite rotar un elemento dándole un ángulo de giro en grados.
- **Skew.** Skew te permite desplazar los ejes horizontales.
- **Scale.** Scale te permite escalar un elemento, toma valores positivos y negativos y se le pueden poner decimales.
- **Translate.** Translate nos permite trasladar un elemento a la vez en el eje de las X y de las Y, dándole las coordenadas iniciales y finales.

#### 5.1.2.2 Ejemplo

En este ejemplo de transición es aplicado a las imágenes; lo que haremos es que se produzca una transformación aumentando la escala de las imágenes 50% más cuando el usuario pase el mouse por encima de la imagen,

Como primer paso definiremos las propiedades de la transición:

- La transición debe aplicarse a todos los elementos que cambien: all.
- La transición dure 2 segundos (transition-duration: 1s).
- La transición se produzca de forma lenta al principio y acabe lenta (transition-timing-function: ease).

```
img.growth {
    width:150px;
    height:130px;
    border:1px solid;

    -webkit-transition:all 2s ease-in-out;
    -moz-transition:all 2s ease-in-out;
    transition: all 2s ease-in-out;
}
```

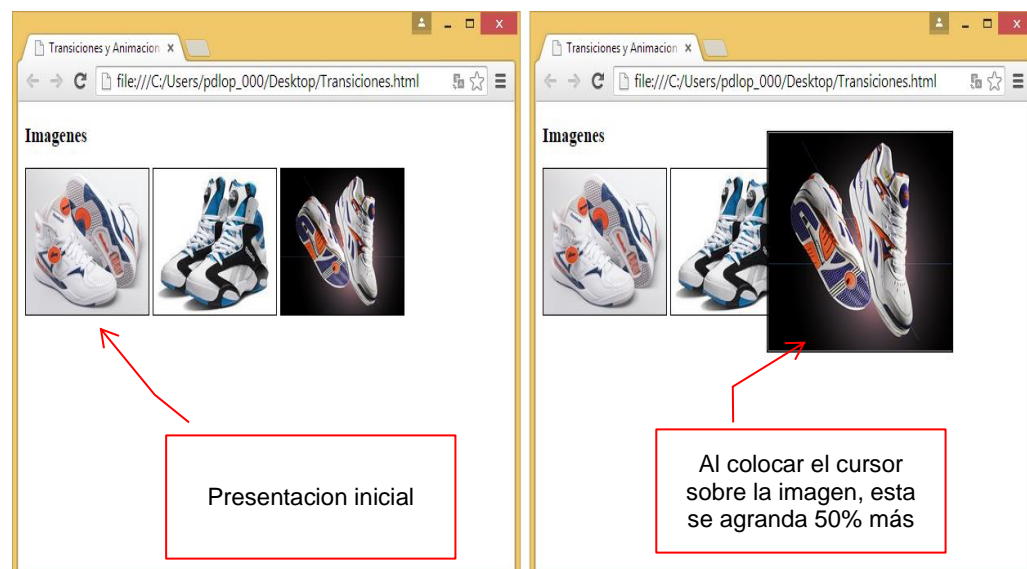
A continuación, definimos las propiedades del evento hover de la imagen, donde al pasar el cursor sobre la imagen, esta aumentara su tamaño 50% mas.

```
img.growth:hover {  
    transform: scale(1.5,1.5);  
}
```

En la página, primero defina la etiqueta <img> definiendo la clase growth, para aplicar el zoom a la imagen.



Al finalizar el diseño de la página, ejecutarla en un navegador, Chrome. Al inicio se visualiza como la pagina de la izquierda; pero al colocar el cursor sobre una imagen, se muestra como la pagina de la derecha.



## Resumen

- 📖 CSS3 pretende estandarizar dos nuevas características: las transiciones y las animaciones que actúan sobre el estado de un elemento transformándolo de manera gradual. Con esto se consiguen efectos muy agradables que mejoran notablemente la experiencia del usuario al interactuar con la página.
- 📖 Las transiciones es el efecto de un cambio en el estado de un elemento de forma gradual en un intervalo de tiempo. Cuando hablamos de cambio de estado nos referimos a cambios en los valores de sus propiedades. Las transiciones únicamente son soportadas por los navegadores: Firefox, Chrome, Safari y Opera. Internet Explorer (al menos hasta su versión 9) no las soporta.
- 📖 Para aplicar una transición sobre un determinado elemento, contamos con las siguientes propiedades: transition-property: Determina la(s) propiedad(es) a ser animadas; puede ser cualquier propiedad CSS. transition-duration: Indica la duración de la animación del inicio al fin en segundos. Por defecto el valor es 0. transition-timing-function: Especifica la velocidad que se desarrollará la transición utilizando las llamadas curvas de Bezier. Así podemos programar la transición para que empiece a una velocidad, luego frene y después se acelere o darle movimiento. transition-delay: Tiempo en que la animación debe pausarse antes de comenzar. Por defecto es 0. Transition: permite unificar todas las propiedades anteriores en una sola
- 📖 Las animaciones son bastante parecidas a las transiciones. Ambas tienen en común que actúan sobre los valores de las propiedades de un elemento transformándolos gradualmente en otros valores. Sin embargo, las animaciones ofrecen bastante más control que las transiciones en el proceso de transformación del elemento. Las animaciones son algo distintas ya que se lanzan explícitamente. No se espera a que suceda un cambio en el estado del elemento como pasaba con las transiciones. Es por ello que para usar animaciones necesitamos "keyframes". Estos "keyframes" describirán el comportamiento que tendrá la animación durante su periodo de ejecución.
- 📖 Las transformaciones de CSS3 pueden ser en 2D y en 3D, siendo estas últimas considerablemente más complejas al tener que tener en cuenta los tres ejes de coordenadas.
- 📖 De las transformaciones de CSS3 en 2D, las más usadas son: Rotate. Rotate te permite rotar un elemento dándole un ángulo de giro en grados. Skew: desplazamiento de los ejes horizontales. Scale. Scale te permite escalar un elemento, toma valores positivos y negativos y se le pueden poner decimales. Translate. Translate nos permite trasladar un elemento a la vez en el eje de las X y de las Y, dándole las coordenadas iniciales y finales.
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 <http://www.csslab.cl/2011/01/31/animaciones-y-transformaciones-con-css3/>

🔗 <http://www.desarrolloweb.com/articulos/transiciones-css3.html>

🔗 <http://www.esandra.com/transformaciones-css3-transiciones-animaciones-css3>

🔗 <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=CSS3TransicionesAnimaciones>

sAnimaciones



# FORMULARIOS

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña formularios para el envío de mensajes de los usuarios.

## TEMARIO

### 6.1 Tema 9: Formularios

6.1.1 Elementos clásicos de formulario: form, input, select, button, textarea.

6.1.2 Nuevos atributos para la etiqueta INPUT: number, range, url, email, date, etc.

6.1.3 Atributos del formulario

6.1.3.1 Vistosidad de las entradas: placeholder, autofocus, etc.

6.1.3.2 Validación de las entradas: required, pattern, etc.

### 6.2 Tema 10: Publicación de un sitio web

6.2.1 Uso de un sitio Web para el envío de datos por un formulario

6.2.2 Publicación del proyecto web en un sitio web

## ACTIVIDADES PROPUESTAS

- Los alumnos diseñan formularios para el envío de mensajes
- Los alumnos diseñan un formulario validando el ingreso de datos para el envío de mensajes.



## 6.1 FORMULARIOS

HTML es un lenguaje de marcado cuyo propósito principal consiste en estructurar los contenidos de los documentos y páginas web. Sin embargo, HTML también incluye elementos para crear aplicaciones web. El estándar HTML/XHTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.

Los años transcurridos desde la publicación de los estándares de HTML y XHTML ha provocado que no estén disponibles todos los elementos utilizados por los formularios más avanzados y modernos. No obstante, HTML/XHTML incluye los suficientes elementos de formulario para crear desde los formularios sencillos que utilizan los buscadores hasta los formularios complejos de las aplicaciones más avanzadas.

### 6.1.1 Elementos clásicos de un formulario

Los formularios se pueden crear utilizando solamente dos etiquetas: **<form>** y **<input>**.

La etiqueta **<form>** encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta **<input>** permite definir varios tipos diferentes de elementos (botones y cuadros de texto).

Los elementos de formulario como botones y cuadros de texto también se denominan "campos de formulario" y "controles de formulario". La mayoría de controles se crean con la etiqueta **<input>**, por lo que su definición formal y su lista de atributos es muy extensa:

- **type** = "text | password | checkbox | radio | submit | reset | file | hidden | image | button" - Indica el tipo de control que se incluye en el formulario
- **name** = "texto" - Asigna un nombre al control (es imprescindible para que el servidor pueda procesar el formulario)
- **value** = "texto" - Valor inicial del control
- **size** = "unidad de medida" - Tamaño inicial del control (para los campos de texto y de password se refiere al número de caracteres, en el resto de controles se refiere a su tamaño en pixel)
- **maxlength** = "numero" - Máximo número de caracteres para los controles de texto y de password
- **checked** = "checked" - Para los controles checkbox y radiobutton permite indicar qué opción aparece preseleccionada
- **disabled** = "disabled" - El control aparece deshabilitado y su valor no se envía al servidor junto con el resto de datos
- **readonly** = "readonly" - El contenido del control no se puede modificar
- **src** = "url" - Para el control que permite crear botones con imágenes, indica la URL de la imagen que se emplea como botón de formulario
- **alt** = "texto" - Descripción del control

#### Cuadro de texto

Se trata del elemento más utilizado en los formularios. En el caso más sencillo, se muestra un cuadro de texto vacío en el que el usuario puede escribir cualquier texto. A continuación, se muestra el código HTML correspondiente al ejemplo anterior:

```
<input type="text" name="nombre" value="" />
```

El atributo **type** diferencia a cada uno de los diez controles que se pueden crear con la etiqueta **<input>**. Para los cuadros de texto, su valor es **text**. El atributo **name** es el más importante en los campos del formulario. De hecho, si un campo no incluye el atributo **name**, sus datos no se envían al servidor. El valor que se indica en el atributo **name** es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario.

El atributo **value** se emplea para establecer el valor inicial del cuadro de texto. Si se crea un formulario para insertar datos, los cuadros de texto deberían estar vacíos. Si no se especifica un tamaño, el navegador muestra el cuadro de texto con un tamaño predeterminado. El atributo **size** permite establecer el tamaño, en caracteres, con el que se muestra el cuadro de texto. Su uso es imprescindible en muchos formularios, en los que algunos campos como la dirección deben mostrar más caracteres de lo normal (`<input size="100">`).

Además de controlar el tamaño con el que se muestra un cuadro de texto, también se puede limitar el tamaño del texto introducido. El atributo **maxlength** permite establecer el máximo número de caracteres que el usuario puede introducir en un cuadro de texto. Su uso es imprescindible para campos como el código postal, el número de la Seguridad Social y cualquier otro dato con formato predefinido y limitado.

Por último, el atributo **readonly** permite que el usuario pueda ver los contenidos del cuadro de texto, pero no pueda modificarlos y el atributo **disabled** deshabilita un cuadro de texto de forma que el usuario no pueda modificarlo y además, el navegador no envía sus datos al servidor.

### **Cuadro de contraseña**

La única diferencia entre este control y el cuadro de texto normal es que el texto que el usuario escribe en un cuadro de contraseña no se ve en la pantalla. En su lugar, los navegadores ocultan el texto utilizando asteriscos o círculos, por lo que es ideal para escribir contraseñas y otros datos sensibles.

```
<input type="password" name="contrasena" value="" />
```

Cambiando el valor del atributo **type** por **password** se transforma el cuadro de texto normal en un cuadro de contraseña. Todos los demás atributos se utilizan de la misma forma y tienen el mismo significado.

### **Checkbox**

Los checkbox o "casillas de verificación" son controles de formulario que permiten al usuario seleccionar y deseleccionar opciones individualmente. Aunque en ocasiones se muestran varios checkbox juntos, cada uno de ellos es completamente independiente del resto. Por este motivo, se utilizan cuando el usuario puede activar y desactivar varias opciones relacionadas, pero no excluyentes.

Puestos de trabajo <br/>

```
<input name="ejecutivo" type="checkbox" value="direccion"/> Dirección  
<input name="tecnico" type="checkbox" value="tecnico"/> Técnico  
<input name="empleado" type="checkbox" value="empleado"/> Empleado
```

El valor del atributo **type** para estos controles de formulario es **checkbox**. Como se muestra en el ejemplo anterior, el texto que se encuentra al lado de cada



checkbox no se puede establecer mediante ningún atributo, por lo que es necesario añadirlo manualmente fuera del control del formulario.

El atributo **value**, junto con el valor del atributo **name**, es la información que llega al servidor cuando el usuario envía el formulario.

Si se quiere mostrar un checkbox seleccionado por defecto, se utiliza el atributo **checked**. Si el valor del atributo es checked, el checkbox se muestra seleccionado. En cualquier otro caso, el checkbox permanece sin seleccionar.

### **Radiobutton**

Los controles de tipo radiobutton son similares a los controles de tipo checkbox, pero presentan una diferencia muy importante: son mutuamente excluyentes. Los radiobutton se utilizan cuando el usuario solamente puede escoger una opción entre las distintas opciones relacionadas que se le presentan. Cada vez que se selecciona una opción, automáticamente se deselecta la otra opción que estaba seleccionada.

Sexo <br/>

<input type="radio" name="sx" value="hombre" checked="checked" /> Hombre

<input type="radio" name="sx" value="mujer" /> Mujer

El valor del atributo type para estos controles de formulario es radio. El atributo **name** se emplea para indicar los radiobutton que están relacionados. Por lo tanto, cuando varios radiobutton tienen el mismo valor en su atributo name, el navegador sabe que están relacionados y puede deselectar una opción del grupo de radiobutton cuando se seleccione otra opción.

### **Botón de envío de formulario**

La mayoría de formularios dispone de un botón para enviar al servidor los datos introducidos por el usuario:

<input type="submit" name="buscar" value="Buscar" />

El valor del atributo **type** para este control de formulario es **submit**. El navegador se encarga de enviar automáticamente los datos cuando el usuario pincha sobre este tipo de botón. El valor del atributo value es el texto que muestra el botón. Si no se establece el atributo value, el navegador muestra el texto predefinido Enviar consulta.

### **Botón de reseteo del formulario**

Aunque su uso era muy popular hace unos años, la mayoría de formularios modernos ya no utilizan este tipo de botón. Se trata de un botón especial que borra todos los datos introducidos por el usuario

<input type="reset" name="limpiar" value="Borrar datos del formulario" />

El valor del atributo **type** para este control de formulario es **reset**. Cuando el usuario pulsa este botón, el navegador borra toda la información introducida y muestra el formulario en su estado original.

### **Campos ocultos**

Los campos ocultos se emplean para añadir información oculta en el formulario

<input type="hidden" name="url\_previa" value="/articulo/primer.html" />

El valor del atributo **type** para este control de formulario es **hidden**. Los campos ocultos no se muestran por pantalla, de forma que el usuario desconoce que el formulario los incluye.

### **Botón de imagen**

El aspecto de los botones de formulario se puede personalizar por completo, ya que incluso es posible utilizar una imagen como botón

```
<input type="image" name="enviar" src="accept.png" />
```

El valor del atributo **type** para este control de formulario es **image**. El atributo **src** indica la URL de la imagen que debe mostrar el navegador en lugar del botón normal.

Su principal ventaja es que permite personalizar por completo la estética de los botones y mostrarlos con un aspecto homogéneo en todos los navegadores. El principal inconveniente es que ralentiza la carga del formulario, y que si se quiere modificar su aspecto, es necesario crear una nueva imagen.

### **Botón**

Algunos formularios complejos necesitan botones más avanzados que los de enviar datos (**type="submit"**) y resetear el formulario (**type="reset"**).

```
<input type="button" name="guardar" value="Guardar Cambios" />
```

El valor del atributo **type** para este control de formulario es **button**. Si pruebas a pulsar un botón de este tipo, verás que el navegador no hace nada: no envía los datos al servidor y no borra los datos introducidos. Este tipo de botones sólo son útiles si se utilizan junto con el lenguaje de programación JavaScript.

### **Fieldset y Legend**

La etiqueta **<fieldset>** agrupa campos del formulario y la etiqueta **<legend>** asigna un nombre a cada grupo.

```
<fieldset>
  <legend>Datos personales</legend>
  Nombre <br/>
  <input type="text" name="nombre" value="" />
  <br/>
  Apellidos <br/>
  <input type="text" name="apellidos" value="" />
  <br/>
  DNI <br/>
  <input type="text" name="dni" value="" size="10" maxlength="9" />
</fieldset>
```

La etiqueta **<fieldset>** agrupa todos los controles de formulario a los que encierra. El navegador muestra por defecto un borde resaltado para cada agrupación. La etiqueta **<legend>** se incluye dentro de cada etiqueta **<fieldset>** y establece el título que muestra el navegador para cada agrupación de elementos.

### Área de Texto

Las áreas de texto son útiles cuando se debe introducir una gran cantidad de texto, ya que es mucho más cómodo de introducir que en un campo de texto normal.

```
<textarea id="idarea" name="txtdescripcion" cols="40" rows="5"></textarea>
```

Los atributos más utilizados en las etiquetas **<textarea>** son los que controlan su anchura y altura. La anchura del área de texto se controla mediante el atributo **cols**, que indica las columnas o número de caracteres que se podrán escribir como máximo en cada fila. La altura del área de texto se controla mediante **rows**, que indica directamente las filas de texto que serán visibles.

El principal inconveniente de los elementos **<textarea>** es que el lenguaje HTML no permite limitar el número máximo de caracteres que se pueden introducir. Mientras los elementos **<input type="text">** disponen del atributo **maxlength**, las áreas de texto no disponen de un atributo equivalente, por lo que sólo es posible limitar el número de caracteres mediante su programación con JavaScript.

### Select

La etiqueta **<select>** define la lista y encierra todas las opciones que muestra la lista. Cada una de las opciones de la lista se define mediante una etiqueta **<option>**. El atributo **value** de cada opción es obligatorio, ya que es el dato que se envía al servidor cuando el usuario envía el formulario. Para seleccionar por defecto una opción al mostrar la lista, se añade el atributo **selected** a la opción deseada.

```
<select id="so" name="so">
  <option value="" selected="selected">- selecciona -</option>
  <option value="windows">Windows</option>
  <option value="mac">Mac</option>
  <option value="linux">Linux</option>
  <option value="otro">Otro</option>
</select>
```

### OptGroup

La etiqueta **<optgroup>** permite agrupar opciones relacionadas dentro de una lista desplegable. Su definición formal se muestra a continuación:

El único atributo que suele utilizarse con la etiqueta **<optgroup>** es **label**, que indica el nombre de cada agrupación. Los navegadores muestran de forma destacada el título de cada agrupación, de forma que el usuario pueda localizar más fácilmente la opción deseada

## 6.1.2 Nuevos atributos para la etiqueta INPUT

El elemento **<input>** tiene nuevos valores para el atributo **type**:

- `<input type="email" />`
- `<input type="url" />`
- `<input type="date" />`
- `<input type="time" />`

- `<input type="datetime" />`
- `<input type="month" />`
- `<input type="week" />`
- `<input type="number" />`
- `<input type="range" />`
- `<input type="tel" />`
- `<input type="search" />`
- `<input type="color" />`

### **Campo email**

En cuestión visual el campo de tipo **email** es básicamente una copia de text pero con algunas peculiaridades propias ya que a la hora de usar **required** valida que el contenido del campo tenga el formato de una cuenta de email.

### **Campo Fecha, Mes, Semana, Hora**

Con HTML5 la funcionalidad de fecha y tiempo viene integrada en el navegador en distintos tipos de campo: date, datetime, time, month y week.

El campo tipo **date** sería el más común, nos ofrece una forma directa de mostrar un calendario y seleccionar una fecha con su día, mes y año. En los distintos dispositivos móviles que usamos nos mostrará una ventana o menú nativo del sistema con el selector de fecha.

Los tipos **datetime** y **datetime-local** son parecidos y te muestran un selector para elegir fecha y hora. El campo datetime se establece con fecha y horario en base a UTC y en cambio datetime-local no tiene definida nunca zona horaria.

El tipo **month** nos deja elegir directamente un año y su mes, mientras que **week** solo el número de la semana del año elegido.

Para elegir una hora ahora usaremos **time** que nos da la selección numérica de hora y minuto, incluido la de segundos y fracciones de segundo en algunos navegadores.

### **Campo Número, Sliders: number & range**

Para el manejo de campos numéricos se han introducido dos nuevos tipos **number** y **range**. Estos dos tipos comparten una serie de propiedades para fijar mínimo, máximo y número de pasos.

- `min="1"` define el mínimo valor aceptado por el campo.
- `max="15"` define el máximo valor.
- `step="2"` en base al mínimo define el rango de valores que sube hasta el máximo.

El campo **number** aparece como uno de texto con la singularidad que tiene flechas para sumar o restar a su valor. En cambio, **range** se muestra como un **slider** o campo deslizante que va desde el mínimo al máximo establecido por min y max y saltando de valor en valor según step.

### **Campo de búsqueda**

El campo **search** funciona exactamente como text pero con la peculiaridad del botón de borrado que aparece justo al escribir en el. Algunos navegadores incluyen el historial de búsqueda del propio site en este tipo de campo y los no compatibles simplemente muestran un campo text.

### **Campo para URLs**

Un campo para escribir una dirección web, comúnmente denominada URL. Igual que el tipo anterior el campo **url** es como text y en dispositivos móviles añade a su teclado virtual teclas de acceso rápido como las terminaciones de dominio .com, .net o .org e incluso el último historial de textos introducidos en este tipo de campo.

### **Campo Selector de color**

El tipo **color** genera un campo que te mostrará el selector de paleta de colores según el sistema operativo que este usando, en el caso de Win o Mac o Linux, será el mismo que puedas usar en cualquier app de diseño como Photoshop o Gimp. En dispositivos móviles muestra una ventana por encima de la aplicación que normalmente te deja escoger entre varios colores.

### **Campo Teléfono**

El campo de teléfono **tel** está básicamente pensado para el uso en dispositivos móviles ya que a que nivel de desktops no añade nada especial y hasta ahora la compatibilidad se basa en sus versiones mobile. Este te muestra el teclado virtual con el formato de teléfono para rellenar el campo.

## **6.1.3 Atributos del formulario**

La mayoría de formularios utilizan los atributos **action** y **method**. El atributo **action** indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador. El atributo **method** establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son GET y POST. De esta forma, casi todos los formularios incluyen el atributo `method="get"` o el atributo `method="post"`.

El método POST permite el envío de mucha más información que el método GET. En general, el método GET admite como máximo el envío de unos 500 bytes de información. La otra gran limitación del método GET es que no permite el envío de archivos adjuntos con el formulario. Además, los datos enviados mediante GET se ven en la barra de direcciones del navegador (se añaden al final de la URL de la página), mientras que los datos enviados mediante POST no se pueden ver tan fácilmente. Por su parte, el método POST se debería utilizar cuando el formulario modifica la información original (insertar, modificar o borrar alguna información).

### **6.1.3.1 Vistosidad de las entradas: placeholder, autofocus**

Con la remesa de nuevas entradas también ha llegado diferente tipo de propiedades, estos los principales.

#### **Autofocus**

El navegador fija el foco al campo que lo tenga indicado automáticamente al cargar la página, ideal para sitios donde el campo de búsqueda es importante.

```
<input name="search" type="search" autofocus="" />
```

**Placeholder**

Mostrará un texto por defecto mientras el campo este vacío y al escribir desaparecerá, una mejora que ayuda a entender el propio campo.

```
<input name="search" type="search" placeholder="Buscar..." />
```

**6.1.3.2 Validación de las entradas: required, pattern****Required**

El campo con este atributo será validado por el propio navegador, tanto si está vacío como si incluye un tipo como puede ser email o url aparecerá un pequeño globo informativo indicando el problema.

```
<input name="email" required="" type="email" />
```

**Pattern**

Esta propiedad nos permite usar una expresión regular para que el navegador valide el campo en base a patrón establecido. Debes especificar un title ya que al validar si falla el navegador mostrará ese mismo.

```
<input title="Solo números" pattern="[0-9]*" type="text" />
```

**Min y Max**

Los atributos **min** y **max** del elemento `<input>` restringen los valores máximos y mínimos que pueden ser ingresados.

## LABORATORIO 8.1

Se nos pide diseñar un formulario en una página HTML, para registrar los datos de un contacto. Valide los datos antes de su envío.

### Codificación en HTML

Para diseñar un formulario validado, codifica las etiquetas en la página HTML, tal como se muestra. Para que los input se encuentren validados, utilice los atributos placeholder, required.

```
formulario.html* X
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="utf-8" />
    <title>Formulario de contacto</title>
  </head>
  <body>
    <h1>Formulario de contacto</h1>
    <form method="post">
      <label for="name">Nombre:</label><br />
      <input type="text" id="name" name="name" value="" placeholder="Tu nombre" required="required" autofocus="autofocus" /><p />

      <label for="email">Correo Electronico:</label><br />
      <input type="email" id="email" name="email" value="" placeholder="tu@email.com" required="required" /><p />

      <label for="website">Sitio Web: </label><br />
      <input type="text" id="website" name="website" value="" /><p />

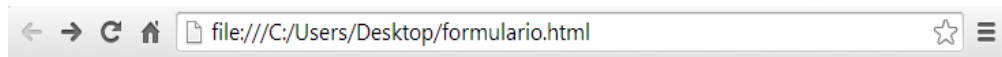
      <label for="subject">Tema: </label><br />
      <select id="subject" name="subject">
        <option value="hola">Solo quiero saludar</option>
        <option value="cuota">Me gustaria una cuota</option>
        <option value="general">General</option>
      </select><p />

      <label for="message">Mensaje:</label><br />
      <textarea name="message" placeholder="Escriba su mensaje aqui." required="required" cols="20" rows="10"></textarea><p />
      <input type="submit" value="Enviar" id="submit" />
    </form>
  </body>
</html>
```

## Ejecutando la página


Al ejecutar la página a través del navegador, se visualiza el formulario y sus controles, tal como se muestra.

En este proceso se puede visualizar el valor del placeholder y cuando no ingresamos datos en un control con el atributo required, se muestra un mensaje.



## Formulario de contacto

Nombre:

Correo Ele  Completa este campo

Sitio Web:

Tema:

Mensaje:



## LABORATORIO 8.2

Se nos pide publicar un formulario para realizar un envío de datos, por MelodySoft, a través de una página HTML.

### Pasos para publicar un formulario

1. Ingrese a la página [www.melodysoft.com](http://www.melodysoft.com)
2. Escriba su login y password.



Después de ingresar el login y clave del usuario creado, verá la siguiente pantalla

3. Haga clic en la opción Formularios de envío.



#### 4. Hacer Clic en la opción Crear nuevo formulario.

Formularios de envío

www.melodysoft.com/panel/form/index.html?visual=02041&action=delete&id=193229

**Melodysoft**

Cerrar sesión 10061966 | Contactar | Regístrate | Panel de control | Publicidad

**Datos personales**

Código: 10061966  
Nombre: damaso  
Fecha de alta: 19/01/2015 15:13  
Última modificación: 19/01/2015 15:13  
Número de foros: 0  
Número de libros: 0  
Número de forms: 8

**Formulario eliminado**

**Formularios de envío**

Localizar por código:  Registros por página: 50

Código	Nombre	Creado
fm1alarcon	formularioaz	27/01/2015
i201419137	frmcontacto	24/01/2015
i201422224	frmcontacto	24/01/2015
i201422334	945176919	27/01/2015
i201422404	frmcontacto	30/01/2015
i201422409	frmcontacto	24/01/2015
i201422415	frmcontacto	31/01/2015
i201422445	frmcontacto	24/01/2015

Formularios: 8 de 8

Haz clic sobre el código de un formulario para modificarlo.

**Crear nuevo formulario**

Volver

Sobre Melodysoft | Aviso legal | Privacidad | Condiciones del servicio | Condiciones de contratación | Índice foros | Tiendas | Boards2

#### 5. Leer las condiciones del servicio y finalmente, active la opción He leído y acepto las condiciones de servicio y haga clic en el botón Aceptar y continuar.

Condiciones particulares de contratación de formularios

www.melodysoft.com/panel/form/tos.html?visual=02041

**Melodysoft**

Cerrar sesión 10061966 | Contactar | Regístrate | Panel de control | Publicidad

**Datos personales**

Código: 10061966  
Nombre: damaso  
Fecha de alta: 19/01/2015 15:13  
Última modificación: 19/01/2015 15:13  
Número de foros: 0  
Número de libros: 0  
Número de forms: 8

**Condiciones particulares**

- Foros de discusión
- Libros de visitas
- Formularios

**Condiciones particulares de contratación de formularios**

Antes de poder crear un formulario en Melodysoft debes leer y aceptar estas condiciones del servicio:

**1.- Objeto**

Las presentes condiciones particulares de uso tienen la finalidad de regular la prestación por parte de MELODYSOFT de su servicio de formularios de envío al CLIENTE ofrecido por Melodysoft S.L. bajo la marca comercial Melodysoft a través de su website www.melodysoft.com, propiedad de Melodysoft S.L., sociedad mercantil constituida por escritura pública otorgada el día 24 de Abril de 2008 ante el Notario D. José Ferrera Almódovar, sociedad inscrita en el Registro Mercantil de Alicante en el Tomo 3304, Folio 109, Hoja 4-113184, Inscrp. 1, con domicilio social en C/ Virgen de la Soledad, 32 de Elda, Alicante - España y provista de C.I.F. nº B-54345830.

Estos formularios ofrecen la posibilidad a terceros de enviar mensajes, vía E-Mail, a los destinatarios especificados por el CLIENTE a través de su panel de control.

Serán CLIENTES del servicio de formularios la persona que actualmente está registrado como tal en el website www.melodysoft.com y que previamente ha aceptado las condiciones generales de uso de los servicios de MELODYSOFT.

**2.- Condiciones particulares**

- Se prohíbe el uso de los formularios de envío de MELODYSOFT con fines contrarios a la buena fe.
- Queda totalmente prohibido el empleo de formularios para enviar contenidos que resulten contrarios a las leyes españolas o que infrinjan los derechos de terceros y, en particular:
  - Cualquier contenido que resulte violento, obsceno, abusivo, ilegal, racial, xenófobo, difamatorio o discriminatorio.
  - Los cráclis, números de serie de programas o cualquier otro contenido que vulnere derechos de propiedad intelectual.
  - Apología del terrorismo y/o genocidio.
  - Pornografía infantil.
  - La publicación de datos personales contraviniendo lo dispuesto en la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.
- Se prohíbe utilizar el formulario contratado en MELODYSOFT para el envío masivo de información no solicitada (SPAM).
- Queda totalmente prohibido, eliminar, ocultar o alterar la publicidad insertada por MELODYSOFT en los formularios de envío, o en sus páginas de notificación de mensajes.
- EL CLIENTE sólo podrá especificar destinatarios del formulario previo consentimiento de éstos para la recepción de mensajes.

☒ **He leído y acepto las condiciones del servicio**

**Aceptar y continuar**

6. Llenar los campos que se solicitan, al finalizar presionar el botón CREAR.

» **Código identificativo formulario**  
fr0001  
El identificador servirá como código de identificación del formulario. No debe contener espacios ni signos extraños. ← Volver

» **Nombre del formulario**  
form1

» **Asunto del formulario**  
Formulario de Envío  
Es el asunto que figurará en el envío del formulario.

» **Dirección E-Mail destino**  
pdlopez@cibertec.edu.pe  
Dirección de E-Mail principal a la que se enviará el formulario.  
Puedes indicarla de estas dos maneras: correo@dominio.com o bien: "mi nombre" <correo@dominio.com>

» **Formato del E-Mail**  
HTML y texto ▼

» **Temática**  
Educación ▼  
Temática de la página del formulario.

» **Campo donde figura el nombre del usuario**  
txtusuario  
Indica el nombre del campo de donde se tomará el nombre del usuario para enviarle la respuesta personalizada y/o para que figure en el campo "From" del formulario.  
Cuando se envía el formulario, o una posible respuesta automática al usuario, en esos E-Mails figura su nombre, este dato se obtiene de los propios datos del formulario, por lo que hay que indicar aquí qué campo es el que contiene el nombre del usuario, si por ejemplo ponemos *nombre* en este campo, entonces la caja de texto donde el usuario indicaría su nombre se llamaría *nombre*:  
<input type="text" name="nombre">

» **Campo donde figura el E-Mail del usuario**  
txtemail  
Indica el nombre del campo de donde se obtiene el E-Mail del usuario para enviarle la respuesta personalizada y/o para que figure en el campo "From" del formulario.  
Cuando se envía el formulario, o una posible respuesta automática, se indica el E-Mail origen (formulario), o se envía al E-Mail del usuario (respuesta), este dato se obtiene de los propios datos del formulario, por lo que hay que indicar aquí qué campo es el que contiene el E-Mail, si por ejemplo ponemos *email* en este campo, entonces la caja de texto donde el usuario indicaría su E-Mail se llamaría *email*:  
<input type="text" name="email">

Crear

← Volver

Hacer click para crear el formulario

Creado el formulario, la plataforma nos indica que el formulario ha sido creado con éxito.



En el formulario de la página HTML, registre el formulario de envío, tal como se muestra.

### La Etiqueta form:

```
<form name="form1" method="post" action="http://forms.melodysoft.com" >
<input name="id" type="hidden" value="nombre_formulario">
```

### Nota

El **nombre\_formulario** es el código identificativo del formulario que se creó en la página de MelodySoft

# Resumen

- 📖 HTML es un lenguaje de marcado cuyo propósito principal consiste en estructurar los contenidos de los documentos y páginas web. Sin embargo, HTML también incluye elementos para crear aplicaciones web. El estándar HTML/XHTML permite crear formularios para que los usuarios interactúen con las aplicaciones web
- 📖 Los formularios se pueden crear utilizando solamente dos etiquetas: **<form>** y **<input>**. La etiqueta **<form>** encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta **<input>** permite definir varios tipos diferentes de elementos (botones y cuadros de texto).
- 📖 El elemento **<input>** tiene nuevos valores para el atributo **type**:
  - `<input type="email" />`
  - `<input type="url" />`
  - `<input type="date" />`
  - `<input type="time" />`
  - `<input type="datetime" />`
  - `<input type="month" />`
  - `<input type="week" />`
  - `<input type="number" />`
  - `<input type="range" />`
  - `<input type="tel" />`
  - `<input type="search" />`
  - `<input type="color" />`
- 📖 La mayoría de formularios utilizan los atributos **action** y **method**. El atributo **action** indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador. El atributo **method** establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son GET y POST. De esta forma, casi todos los formularios incluyen el atributo `method="get"` o el atributo `method="post"`
- 📖 Con la remesa de nuevas entradas también ha llegado diferente tipo de propiedades: **Autofocus**, el navegador fija el foco al campo que lo tenga indicado automáticamente al cargar la página, ideal para sitios donde el campo de búsqueda es importante. **Placeholder**, muestra un texto por defecto mientras el campo este vacío y al escribir desaparecerá.
- 📖 Para validar las entradas en un formulario, tenemos `required`, `pattern`.
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.
  - 🔗 <http://octuweb.com/formularios-html5>
  - 🔗 [https://developer.mozilla.org/es/docs/HTML/HTML5/Forms\\_in\\_HTML5](https://developer.mozilla.org/es/docs/HTML/HTML5/Forms_in_HTML5)
  - 🔗 [http://librosweb.es/libro/xhtml/capitulo\\_8.html](http://librosweb.es/libro/xhtml/capitulo_8.html)
  - 🔗 <http://www.elwebmaster.com/general/formulario-validacion-html5>
  - 🔗 <http://www.genbetadev.com/desarrollo-web/introduccion-a-los-formularios-de-html5-ii>





# FORMULARIOS

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña formularios para el envío de mensajes de los usuarios.

## TEMARIO

### 6.1 Tema 9: Formularios

6.1.1 Elementos clásicos de formulario: form, input, select, button, textarea.

6.1.2 Nuevos atributos para la etiqueta INPUT: number, range, url, email, date, etc.

6.1.3 Atributos del formulario

6.1.3.1 Vistosidad de las entradas: placeholder, autofocus, etc.

6.1.3.2 Validación de las entradas: required, pattern, etc.

### 6.2 Tema 10: Publicación de un sitio web

6.2.1 Uso de un sitio Web para el envío de datos por un formulario

6.2.2 Publicación del proyecto web en un sitio web

## ACTIVIDADES PROPUESTAS

- Los alumnos diseñan formularios para el envío de mensajes
- Los alumnos diseñan un formulario validando el ingreso de datos para el envío de mensajes.





## 6.2 Publicación de un sitio Web

### 6.2.1 Uso de un sitio Web para el envío de datos por un formulario

#### Pasos para publicar un formulario

1. Ingrese a la página [www.melodysoft.com](http://www.melodysoft.com)
2. Escriba su login y password



Después de ingresar el login y clave del usuario creado, verá la siguiente pantalla.

3. Haga clic en la opción Formularios de envío.



#### 4. Hacer clic en la opción Crear nuevo formulario.

Formularios de envío

www.melodysoft.com/panel/form/index.html?visual=02041&action=delete&id=193229

**Melodysoft**

Cerrar sesión 10061966 Contactar Regístrate Panel de control Publicidad

Datos personales...

Código 10061966

Nombre damaso

Fecha de alta 19/01/2015 15:13

Última modificación 19/01/2015 15:13

Número de foros: 0

Número de libros: 0

Número de forms: 8

Condiciones particulares:

- Foros de discusión
- Libros de visitas
- Formularios

Formulario eliminado

Formularios de envío

Localizar por código: [ ] Registros por página: 50

Código	Nombre	Creado
fm1alarcon	formularioaz	27/01/2015
i201419137	frmcontacto	24/01/2015
i201422224	frmcontacto	24/01/2015
i201422334	945176919	27/01/2015
i201422404	frmcontacto	30/01/2015
i201422409	frmcontacto	24/01/2015
i201422415	frmcontacto	31/01/2015
i201422445	frmcontacto	24/01/2015

Formularios 1 - 8 de 8

Haz clic sobre el código de un formulario para modificarlo.

Crear nuevo formulario

Volver

Sobre Melodysoft Aviso legal Privacidad Condiciones del servicio Condiciones de contratación Índice foros Tiendas Boards2

#### 5. Leer las condiciones del servicio y finalmente, active la opción He leído y acepto las condiciones de servicio y haga clic en el botón Aceptar y continuar.

Condiciones particulares de contratación de formularios

Antes de poder crear un formulario en Melodysoft debes leer y aceptar estas condiciones del servicio:

1.- Objeto

Las presentes condiciones particulares de uso tienen la finalidad de regular la prestación por parte de MELODYSOFT de su servicio de formularios de envío al CLIENTE ofrecido por Melodysoft S.L. bajo la marca comercial Melodysoft a través de su website www.melodysoft.com, propiedad de Melodysoft S.L., sociedad mercantil constituida por escritura pública otorgada el día 24 de Abril de 2008 ante el Notario D. José Ferrera Almudover, sociedad inscrita en el Registro Mercantil de Alicante en el Tomo 3304, Folio 109, Hoja 4-113184, Inscrp. 1, con domicilio social en C/ Virgen de la Soledad, 32 de Elda, Alicante - España y provista de C.I.F. nº B-54345830.

Estos formularios ofrecen la posibilidad a terceros de enviar mensajes, vía E-Mail, a los destinatarios especificados por el CLIENTE a través de su panel de control.

Serán CLIENTES del servicio de formularios la persona que actualmente está registrado como tal en el website www.melodysoft.com y que previamente ha aceptado las condiciones generales de uso de los servicios de MELODYSOFT.

2.- Condiciones particulares

- Se prohíbe el uso de los formularios de envío de MELODYSOFT con fines contrarios a la buena fe.
- Queda totalmente prohibido el empleo de formularios para enviar contenidos que resulten contrarios a las leyes españolas o que infrinjan los derechos de terceros y, en particular:
  - Cualquier contenido que resulte violento, obsceno, abusivo, ilegal, racial, xenófobo, difamatorio o discriminatorio.
  - Los cráclis, números de serie de programas o cualquier otro contenido que vulnere derechos de propiedad intelectual.
  - Apología del terrorismo y/o genocidio.
  - Pornografía infantil.
  - La publicación de datos personales contraviniendo lo dispuesto en la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.
- Se prohíbe utilizar el formulario contratado en MELODYSOFT para el envío masivo de información no solicitada (SPAM).
- Queda totalmente prohibido, eliminar, ocultar o alterar la publicidad insertada por MELODYSOFT en los formularios de envío, o en sus páginas de notificación de mensajes.
- EL CLIENTE sólo podrá especificar destinatarios del formulario previo consentimiento de éstos para la recepción de los mensajes.

☒ He leído y acepto las condiciones del servicio

Aceptar y continuar

6. Llenar los campos que se solicitan, al finalizar presionar el botón CREAR.

✦ **Código identificativo formulario**  
fr0001  
El identificador servirá como código de identificación del formulario. No debe contener espacios ni signos extraños. [← Volver](#)

✦ **Nombre del formulario**  
form1

✦ **Asunto del formulario**  
Formulario de Envío  
Es el asunto que figurará en el envío del formulario.

✦ **Dirección E-Mail destino**  
pdlopez@cibertec.edu.pe  
Dirección de E-Mail principal a la que se enviará el formulario.  
Puedes indicarla de estas dos maneras: correo@dominio.com o bien: "mi nombre" <correo@dominio.com>

✦ **Formato del E-Mail**  
HTML y texto ▼

✦ **Temática**  
Educación ▼  
Temática de la página del formulario.

✦ **Campo donde figura el nombre del usuario**  
txtusuario  
Indica el nombre del campo de donde se tomará el nombre del usuario para enviarle la respuesta personalizada y/o para que figure en el campo "From" del formulario.  
Cuando se envía el formulario, o una posible respuesta automática al usuario, en esos E-Mails figura su nombre, este dato se obtiene de los propios datos del formulario, por lo que hay que indicar aquí qué campo es el que contiene el nombre del usuario, si por ejemplo ponemos *nombre* en este campo, entonces la caja de texto donde el usuario indicaría su nombre se llamaría *nombre*:  
<input type="text" name="nombre">

✦ **Campo donde figura el E-Mail del usuario**  
txtemail  
Indica el nombre del campo de donde se obtiene el E-Mail del usuario para enviarle la respuesta personalizada y/o para que figure en el campo "From" del formulario.  
Cuando se envía el formulario, o una posible respuesta automática, se indica el E-Mail origen (formulario), o se envía al E-Mail del usuario (respuesta), este dato se obtiene de los propios datos del formulario, por lo que hay que indicar aquí qué campo es el que contiene el E-Mail, si por ejemplo ponemos *email* en este campo, entonces la caja de texto donde el usuario indicaría su E-Mail se llamaría *email*:  
<input type="text" name="email">

[← Volver](#)

Crear

Hacer click para crear el formulario

Creado el formulario, la plataforma nos indica que el formulario ha sido creado con éxito.



En el formulario de la página HTML, registre el formulario de envío, tal como se muestra.

### La Etiqueta form:

```
<form name="form1" method="post" action="http://forms.melodysoft.com" >
<input name="id" type="hidden" value="nombre_formulario">
```

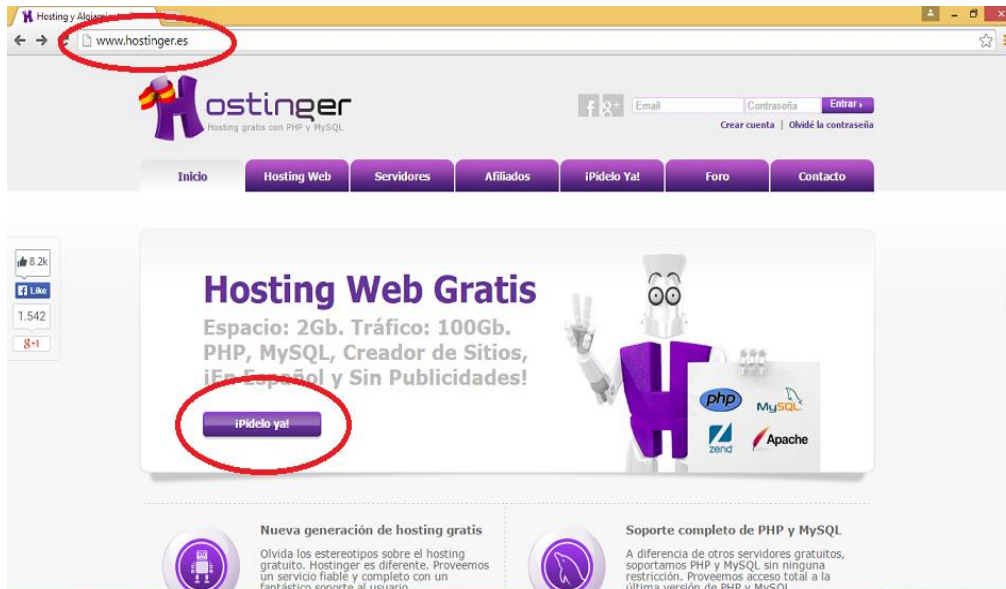
### Nota

El **nombre\_formulario** es el código identificativo del formulario que se creó en la página de MelodySoft

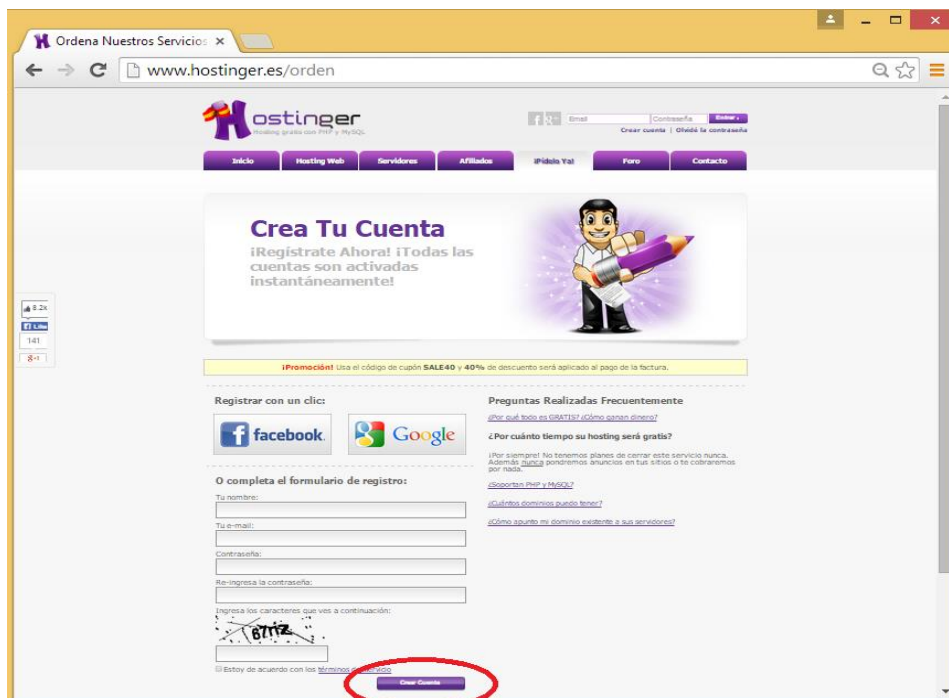
## 6.2.2 Publicación del proyecto web en un sitio web

### Alojamiento Web Gratuito

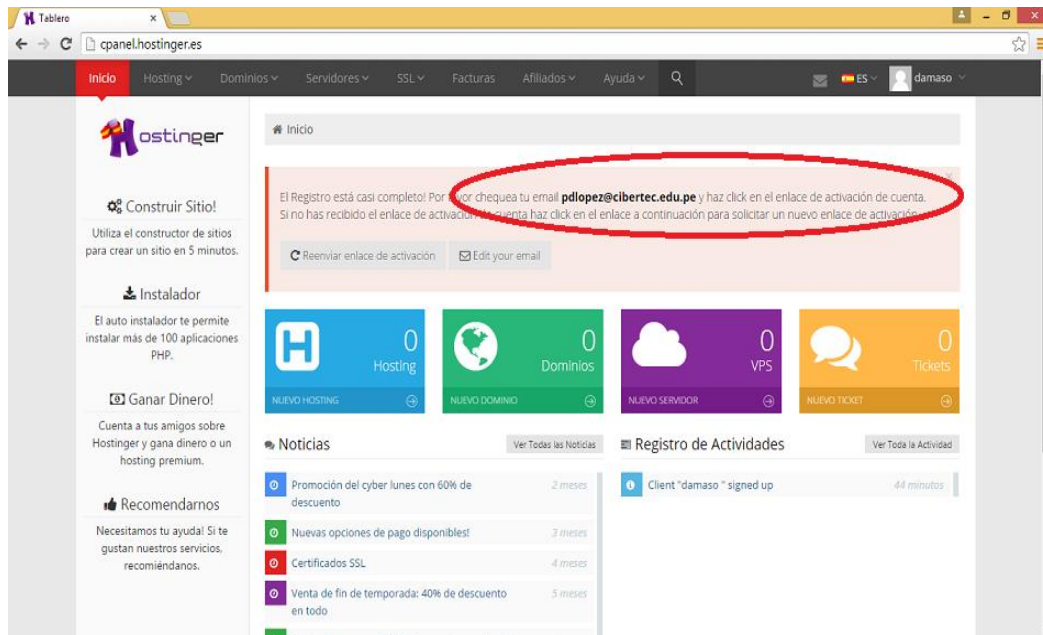
Para alojar tu proyecto Web en forma gratuita ir: [www.hostinger.es](http://www.hostinger.es), y selecciona la opción Hosting Web Gratis, tal como se muestra en la figura.



A continuación, registra tus datos personales, tal como se muestra y presiona el botón CREAR CUENTA.



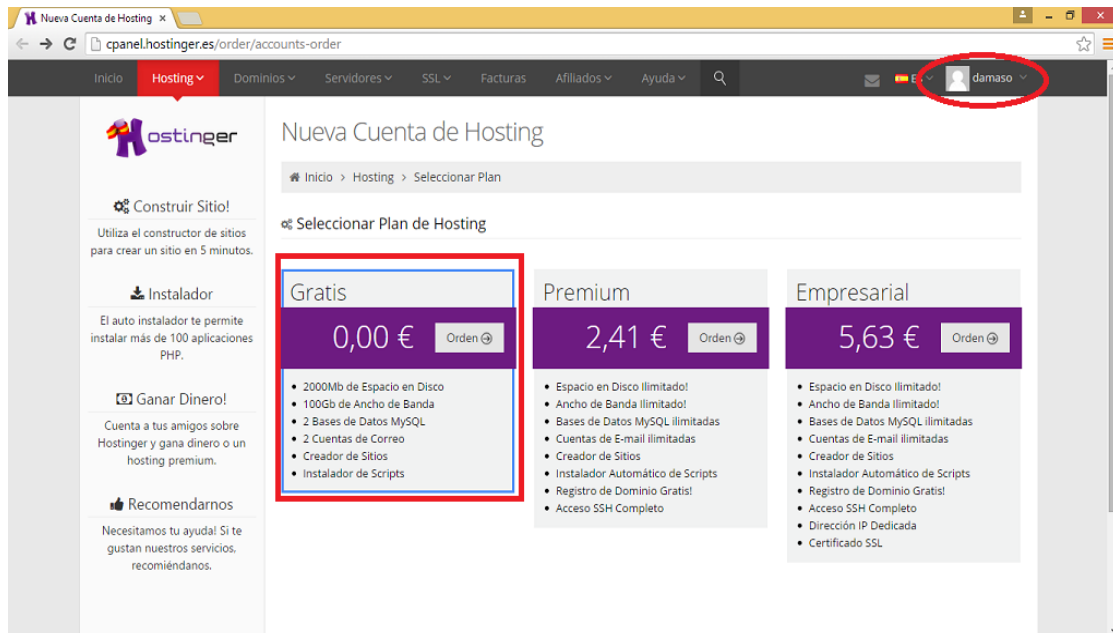
A continuación, el tablero solicitará que actives tu cuenta desde su cuenta de correo, tal como se muestra.



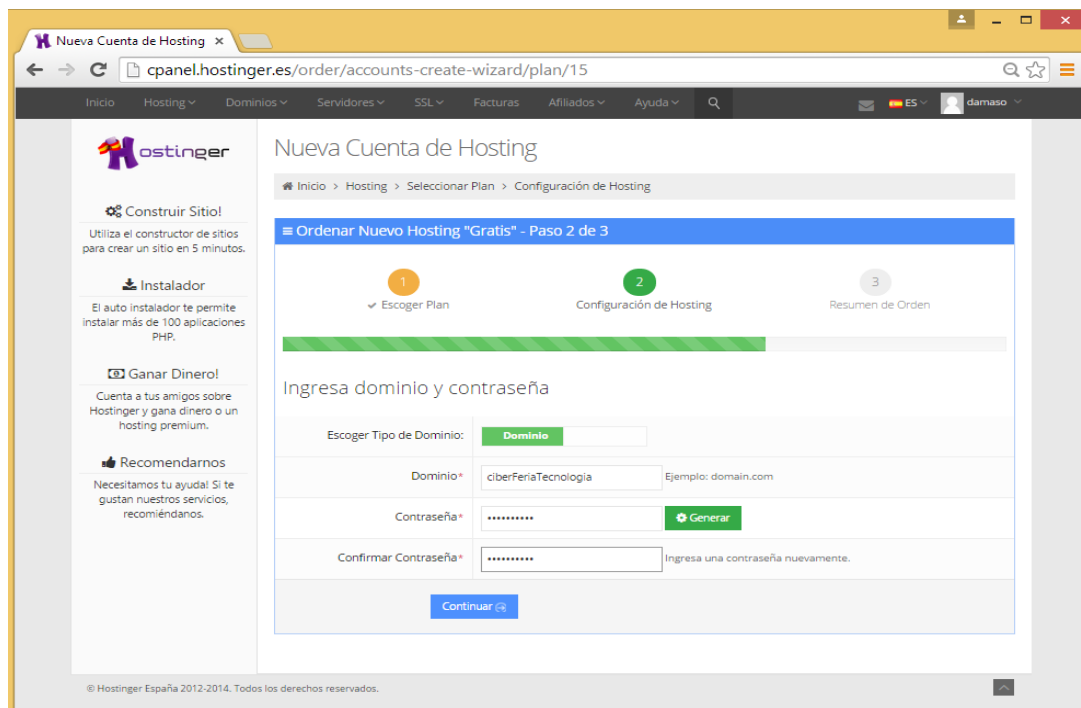
Dirigirse a su cuenta de correo y activar su cuenta, tal como se muestra en la figura



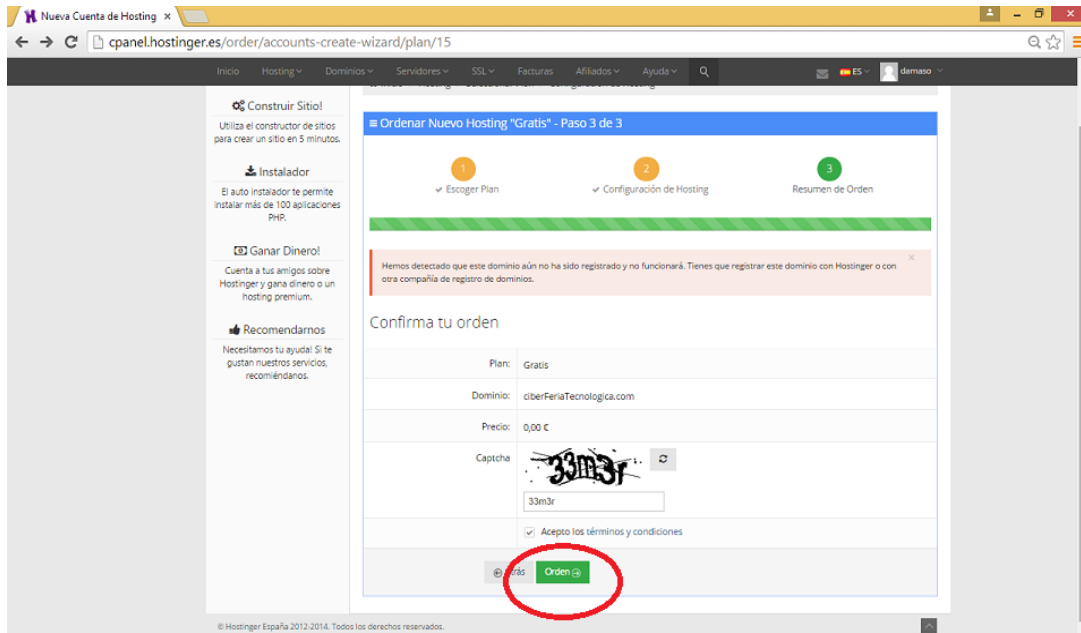
Activada la cuenta, observamos que se visualiza nuestro nombre, selecciona el plan de Hosting; Gratis, tal como se muestra.



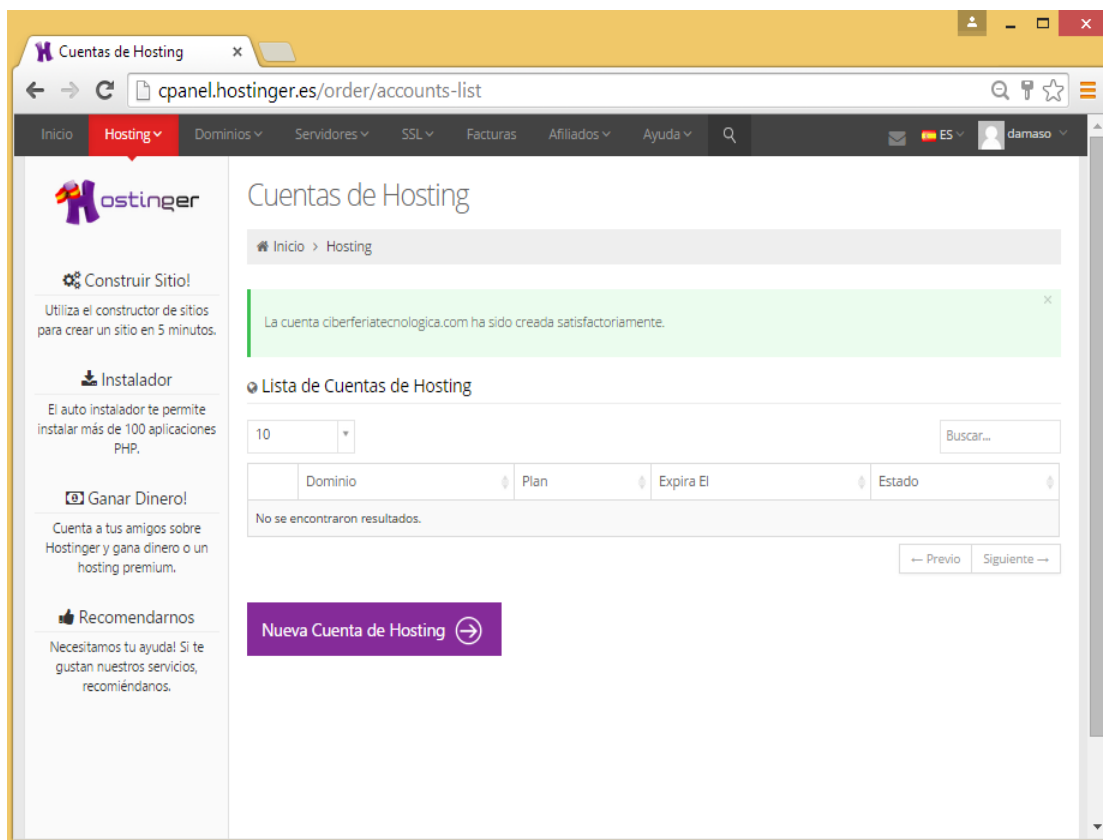
Defina la nueva cuenta de Hosting: ingrese el nombre del dominio: ciberFeriaTecnologica.com, su clave y confirmar. Presiona el botón CONTINUAR.



A continuación, debe confirmar su orden, ingresa el campo captcha, aceptar los términos y presionar el botón ORDEN, tal como se muestra.



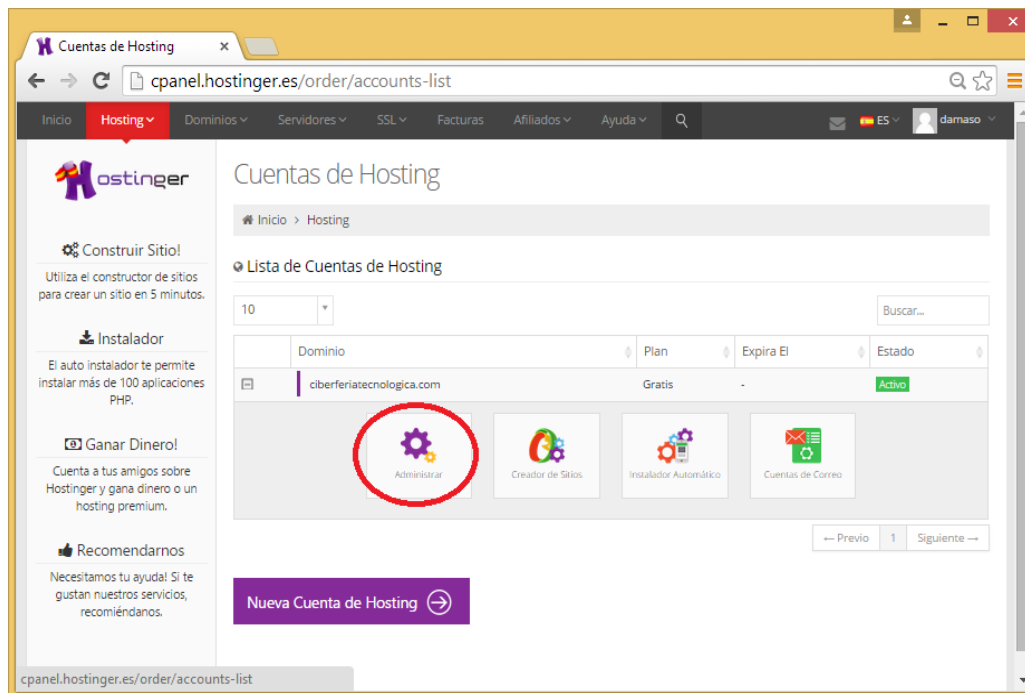
Al finalizar, la cuenta se habrá creado.



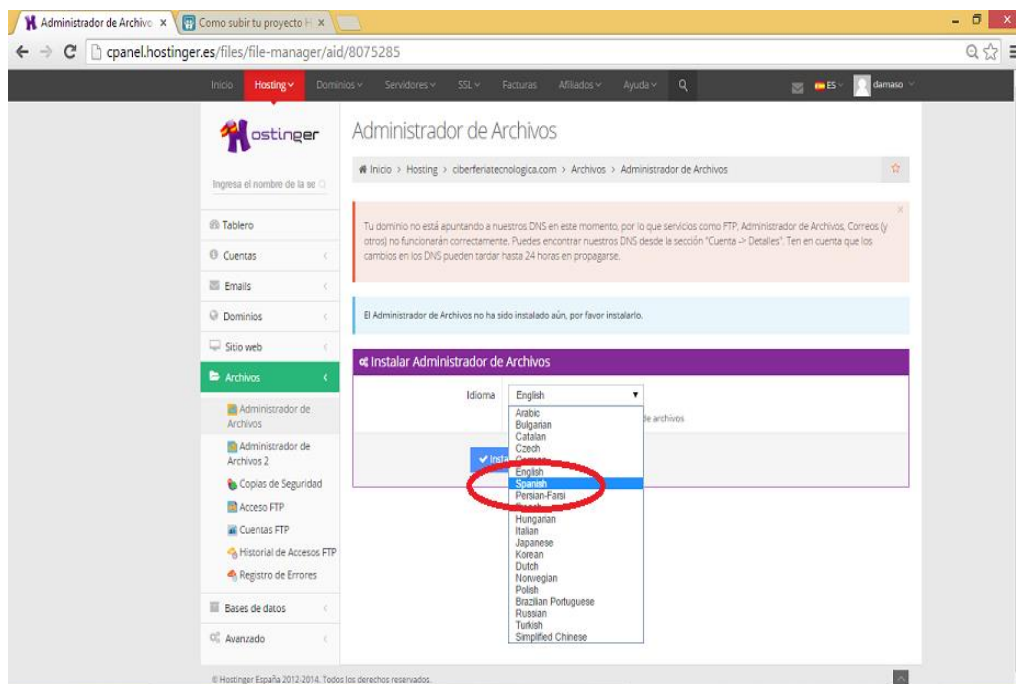


## Manejando un Sitio Web

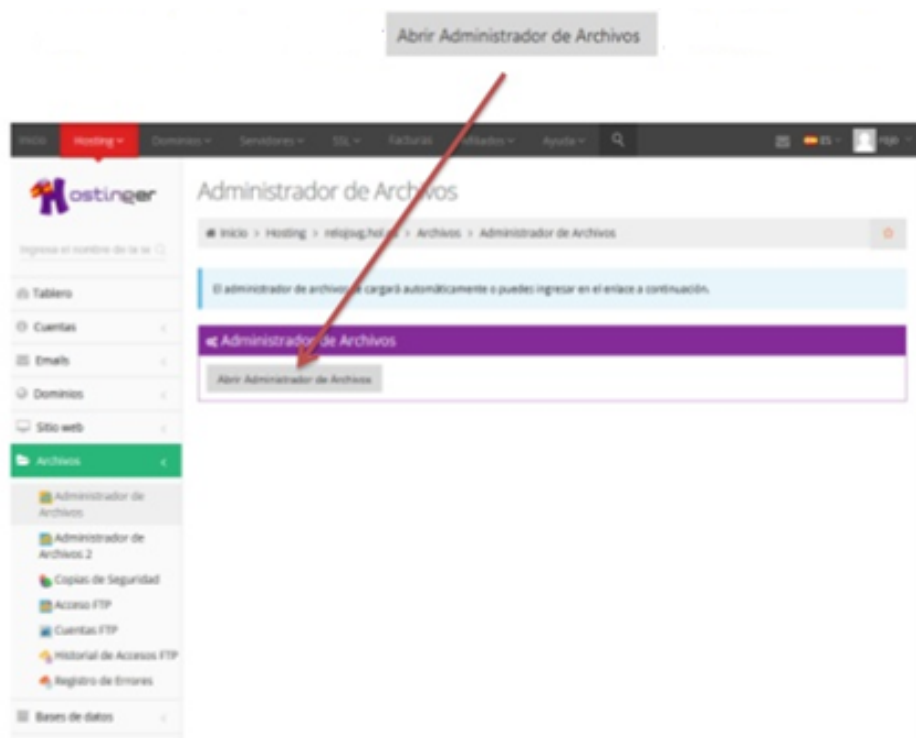
Al seleccionar el dominio creado, se muestra cuatro opciones, tal como se muestra. Presiona el botón Administrar para administrar los archivos de tu proyecto.



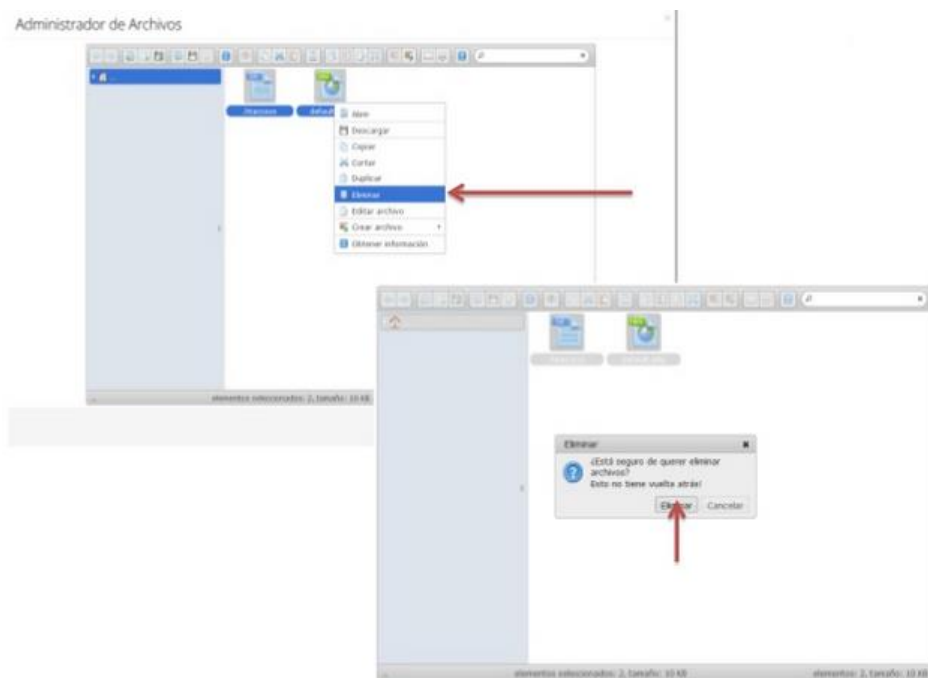
Por ser la primera vez, cargamos un archivo que nos pedirá que instalemos el cargador de archivos, selecciona el idioma, presiona el botón EJECUTAR.



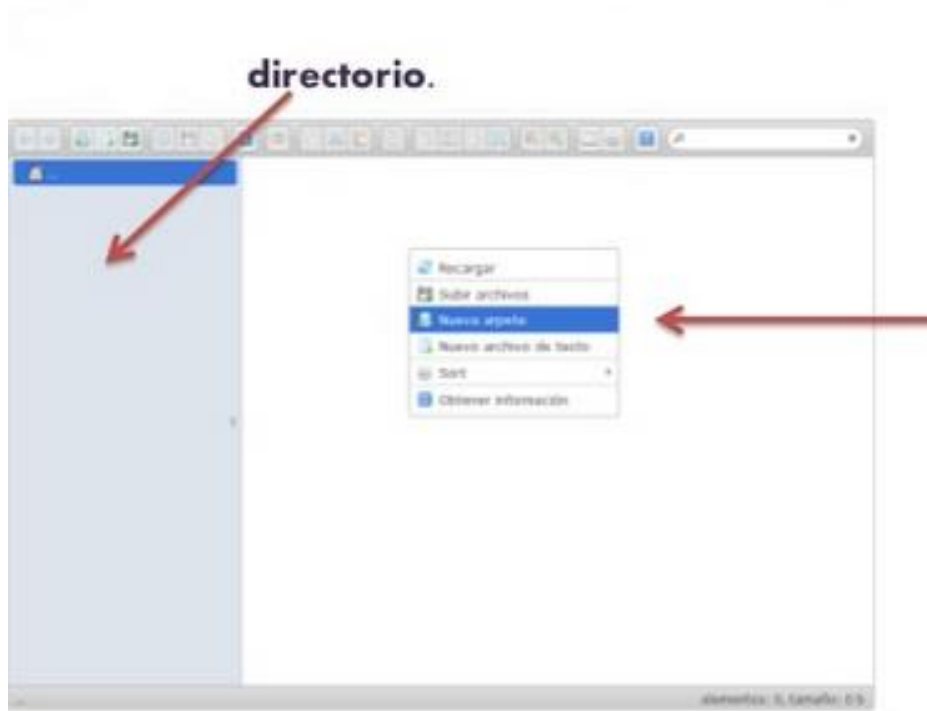
Ya dentro del Administrador de Archivos, si es la primera vez que cargamos un archivo, nos pedirá que instalemos el cargador e archivos, lo instalan y listo.



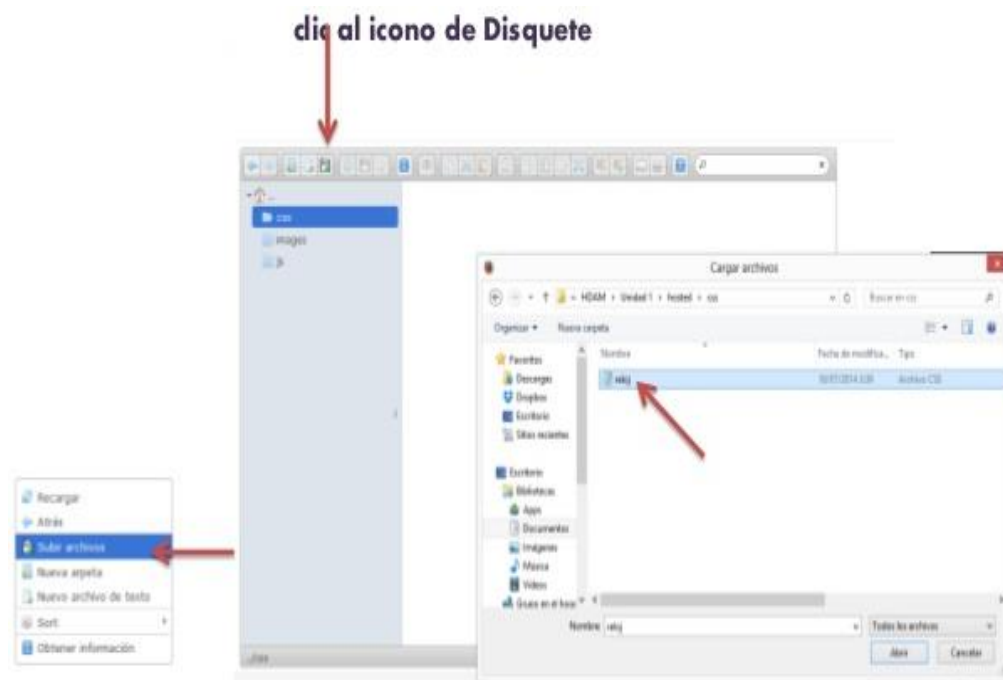
Por defecto aparecen dos archivos, eliminarlos.



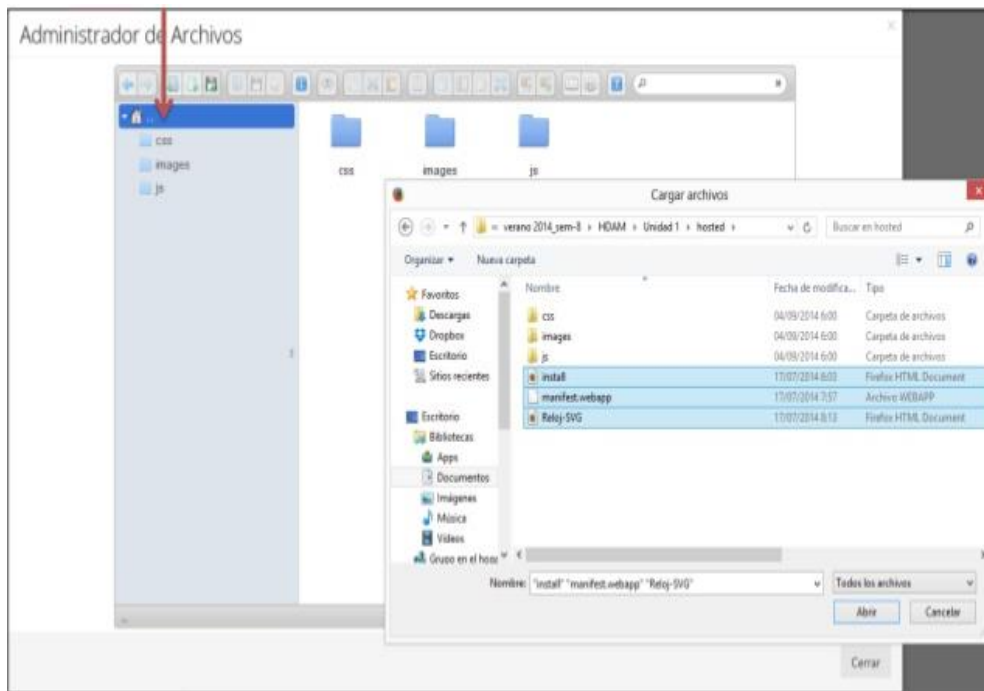
Antes de proceder a agregar archivos, primero creamos directorios para guardar los archivos según su tipo.



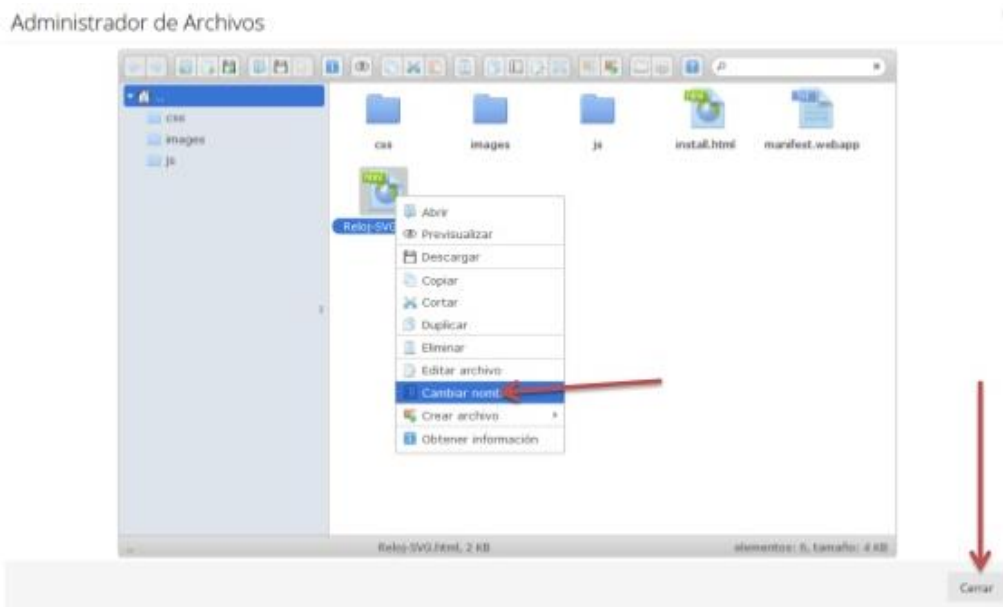
Creando las carpetas, subimos los archivos para cada carpeta. Selecciona una carpeta, hacer clic al icono Disquete para cargar archivos al directorio.



Al concluir de subir los archivos, nos dirigimos al directorio raíz y agregamos los archivos que faltan.



Cargado los archivos, debemos asignar a la pagina principal como index.html debido a que el hosting buscara el index como pagina de inicio. Luego cerramos.



También puede consultar: <http://es.slideshare.net/israfenix/como-subir-tuhostinger>